

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Modalités d'interaction pour les personnes lourdement handicapées: étude et conception d'une application de téléphonie

Rousseau, Pierre

*Award date:*  
2017

*Awarding institution:*  
Université de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

UNIVERSITÉ DE NAMUR  
Faculté d'informatique  
Année académique 2016–2017

**Modalités d'interaction pour les  
personnes lourdement handicapées :  
étude et conception d'une  
application de téléphonie**

Pierre ROUSSEAU



Maîtres de stage : Sébastien ANNYS & François VANDER LINDEN

Promoteur : \_\_\_\_\_ (Signature pour approbation du dépôt - REE art. 40)  
M. DUMAS

Mémoire présenté en vue de l'obtention du grade de  
Master en Sciences Informatiques.



# Résumé - Abstract

A l'aide de l'informatique, la vie quotidienne des personnes lourdement handicapées s'est considérablement améliorée ces dernières années. Elles peuvent retrouver une certaine forme d'autonomie, en utilisant notamment le joystick d'une chaise roulante ou encore une télécommande. L'objectif de ce travail est double : premièrement, pouvoir proposer une solution pour téléphoner au travers d'une interface adaptée, et deuxièmement proposer une aide à la décision lorsqu'il s'agit de sélectionner une modalité d'interaction en fonction du handicap d'une personne. Ces objectifs ont été remplis en concevant une application VoIP adaptée ainsi qu'un arbre de décision des modalités d'interaction. Cela permettra d'améliorer encore un peu plus le quotidien des personnes lourdement handicapées, qui ont la possibilité d'avoir une solution de téléphonie adaptée à leurs besoins et leurs difficultés.

**Mots-clés :** Interaction, modalités, personnes lourdement handicapées, VoIP

With the help of computer science, the daily life of severely disabled people has improved greatly in the recent years. They are able to regain autonomy, using for instance the joystick of a wheelchair or a remote control. The goal of this work is twofold : first, to be able to propose a solution to make phone calls through an adapted interface, and second, to propose a decision support when choosing a modality of interaction according to the handicap of the user. These objectives were fulfilled by designing a suitable VoIP application as well as a decision tree for interaction modalities. This will make possible to further improve the daily life of severely disabled people who have the opportunity to have a telephony solution adapted to their needs and difficulties.

**Keywords :** Interaction, modalities, severely disabled people, VoIP



# Avant-propos

Avant de commencer ce mémoire, je souhaite adresser ici tous mes remerciements aux personnes qui ont contribué de près ou de loin à l'élaboration de mon mémoire.

Tout d'abord, je voudrais remercier mon promoteur, M. Dumas, pour son implication, son aide, ses éclaircissements ainsi que ses relectures tout au long de la rédaction de mon mémoire.

Ensuite, je souhaite remercier mes deux maîtres de stage, M. Annys et M. Vander Linden, pour leur accueil au sein de leur entreprise (Home-Based), leur suivi et leurs conseils précieux tant dans la conception que dans le développement de ma solution.

Je voudrais également remercier Jonathan Maes, avec qui j'ai effectué une partie de mon stage dans l'entreprise Home-Based. Nous avons collaboré sur certains points de la conception.

Je remercie aussi les deux ergothérapeutes que nous avons rencontrées au CRETH, ainsi que l'utilisatrice que nous avons eu la chance d'interviewer. Leurs conseils, retours et avis ont été fort utiles tout au long de la conception de ma solution.

Je voudrais également remercier toutes les personnes qui ont apporté une relecture à mon travail.

Enfin, pour tout ce qu'ils m'ont appris, je remercie l'ensemble des professeurs et assistants de la faculté d'informatique que j'ai croisé durant mes cinq années d'études.



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Contexte . . . . .	7
1.2	Home-Based . . . . .	7
1.3	Objectif du travail et méthodologie . . . . .	8
1.4	Collaboration avec Jonathan Maes . . . . .	9
1.5	Structure du travail . . . . .	9
<b>2</b>	<b>Stage : une interface de téléphonie pour personnes handicapées</b>	<b>10</b>
2.1	Objectifs de l'application . . . . .	10
2.2	Méthodologie . . . . .	11
2.3	Collecte d'informations auprès d'experts . . . . .	11
2.4	Solutions existantes . . . . .	13
2.4.1	Solutions VoIP . . . . .	13
2.4.2	Bibliothèques graphiques . . . . .	16
2.5	Méthodes VoIP . . . . .	16
2.6	Données . . . . .	20
2.7	Interaction . . . . .	22
2.8	Interface . . . . .	25
2.8.1	Gestion des appels . . . . .	26
2.8.2	Gestion des contacts . . . . .	28
2.8.3	Gestion des historiques d'appels . . . . .	31
2.8.4	Fermeture de l'application . . . . .	33
2.8.5	Boîtes de dialogue . . . . .	33
2.9	Validation utilisateur . . . . .	35
2.9.1	Utilisateurs . . . . .	35
2.9.2	Méthodologie . . . . .	35
2.9.3	Observation . . . . .	35
2.9.4	Ressenti de l'utilisatrice et avis de l'ergothérapeute . . . . .	36
2.10	Conclusion . . . . .	36
<b>3</b>	<b>Regard critique sur le stage</b>	<b>37</b>
3.1	Liblinphone et Qt . . . . .	37
3.2	Gestion des données . . . . .	37
3.3	Interaction et interface . . . . .	38
3.4	Objectifs . . . . .	38
3.5	Conclusion . . . . .	39



<b>4</b>	<b>État de l’art</b>	<b>40</b>
4.1	Modalités d’interaction d’entrée . . . . .	41
4.1.1	Mécanique . . . . .	41
4.1.2	Parole . . . . .	49
4.1.3	Regard . . . . .	49
4.1.4	Signaux bio-électriques . . . . .	50
4.2	Balayage d’éléments . . . . .	50
4.2.1	Techniques . . . . .	50
4.2.2	Applications . . . . .	55
4.3	Entrée de texte . . . . .	57
4.3.1	Sélection des caractères . . . . .	58
4.3.2	Arrangement des caractères . . . . .	60
4.4	Conclusion . . . . .	61
<b>5</b>	<b>Amélioration de l’interface de téléphonie</b>	<b>62</b>
5.1	Amélioration des interactions . . . . .	63
5.2	Gestion des contacts . . . . .	63
5.3	Architecture globale . . . . .	63
5.4	Détails d’implémentation . . . . .	65
5.4.1	Partie interactions . . . . .	65
5.4.2	Partie VoIP . . . . .	66
5.4.3	Partie contacts . . . . .	66
5.5	Analyse des améliorations réalisées . . . . .	67
<b>6</b>	<b>Élaboration d’une taxonomie des modalités d’interaction</b>	<b>68</b>
<b>7</b>	<b>Conclusion et travaux futurs</b>	<b>71</b>
7.1	Travaux futurs . . . . .	72

# Glossaire

**CardDAV** : Protocole de gestion d'un carnet d'adresse.

**DTMF - Dual-Tone Multi-Frequency** : Couple de fréquence utilisé dans la téléphonie pour interagir notamment avec des serveurs vocaux.

**Modalité d'interaction** : Canal de communication entre un humain et une machine.

**Moyen d'interaction** : Manière d'interagir avec une machine.

**RFC** : Série de document décrivant la partie technique des éléments liés à internet et aux réseaux.

**SIP - Session Initiation Protocol** : Protocole de gestion de sessions multimédia. Le plus courant pour la VoIP.

**TTS - Text-To-Speech** : Système permettant de synthétiser vocalement un texte écrit.

**vCard** : Format d'échange de données personnelles.

**VoIP - Voice over IP** : Technologie permettant de faire passer la voix sur un réseau IP.

**WebSocket** : Protocole réseau permettant la création d'une communication en *full-duplex* à l'aide d'une connexion TCP.



# Chapitre 1

## Introduction

### 1.1 Contexte

Allumer une lampe, ouvrir sa porte d'entrée, changer de chaîne sur sa télévision, répondre au téléphone, toutes ces actions sont banales pour la plupart des gens. Cependant, c'est un vrai challenge pour une personnes à capacité motrice réduite. En effet, sans possibilité de se mouvoir ou de contrôler une télécommande, toutes ces actions sont extrêmement compliquées, voire impossibles. La personne perd donc son autonomie et est contrainte de demander une aide constante.

Ce constat est valable pour toute personne ayant des difficultés motrices, aussi bien les personnes âgées que les personnes handicapées. En Belgique, selon la Direction générale de Statistiques<sup>1</sup>, la population âgée de plus de 67 ans représentera 22.51%[Stabel, 2013] de la population totale en 2060, alors qu'aujourd'hui, elle ne représente que 16.29%. En 1991, ce segment de population correspondait à 12.87% de la population totale. Ces chiffres soulignent une augmentation de près de 100% en soixante-neuf ans. Cette augmentation permet de se rendre compte de l'enjeu de trouver des solutions pour rendre à ce segment de personnes leur autonomie.

### 1.2 Home-Based

C'est pour pouvoir redonner de l'autonomie aux personnes à mobilité réduite que la société Home-Based<sup>2</sup> à été fondée en 2015. Home-Based est une start-up namuroise créée par Sébastien ANNYS et François VANDER LINDEN. Dans ce contexte, ils ont développé SOLINE<sup>3</sup>, un système de contrôle d'environnement adapté. Soline permet de contrôler une habitation (allumer la lumière, contrôler la télévision, etc) en utilisant un mode d'interaction adapté. Ils proposent toute une série de modalités d'interactions, notamment :

- Interrupteur classique
- Télécommande
- Tablette ou smartphone
- Ordinateur personnel
- Commande de fauteuil roulant

---

1. <http://statbel.fgov.be/fr/statistiques/chiffres/>

2. <http://www.home-based.eu/>

3. <http://www.home-based.eu/soline.php>

- Contacteur
- Geste
- Voix

Quiconque est capable d'utiliser au moins un de ces différents modes d'interaction est capable d'utiliser Soline, et ainsi est capable de contrôler son habitation. Ces différents modes peuvent coexister, et donc s'adapter à tous les occupants d'une habitation.

Une des grandes forces d'Home-Based est leur capacité à adapter leur système à toute habitation et tout équipement. Ils peuvent prendre la main sur n'importe quel luminaire, n'importe quelle télévision, etc. Il n'est pas nécessaire d'avoir des appareils domotiques pour pouvoir bénéficier de Soline.

## 1.3 Objectif du travail et méthodologie

Afin d'améliorer encore plus le quotidien des personnes lourdement handicapées, il est possible de se poser la question suivante :

Comment est-il possible d'améliorer la vie quotidienne des personnes lourdement handicapées grâce à l'informatique, en particulier à l'aide des modalités et moyens d'interaction ?

Cette question est assez large et résolue en partie par l'entreprise Home-Based. Cependant, il est possible de dégager deux sous-questions, qui seront abordées dans ce travail :

1. Comment permettre aux personnes lourdement handicapées de pouvoir téléphoner ainsi que d'ouvrir la porte d'entrée via un portique aisément ?
2. Quelles modalités d'interaction sont-elles disponibles en fonction du handicap de la personne ?

Pour apporter une réponse à ces deux questions, la méthodologie suivante a été suivie :

En premier lieu, une immersion a lieu au sein de l'entreprise Home-Based, dans le cadre du stage. L'objectif y est double : premièrement de découvrir le domaine de travail, au travers de plusieurs rencontres avec des experts du domaine ainsi qu'une utilisatrice du système Soline. Deuxièmement, de développer le prototype d'une application VoIP, c'est-à-dire de téléphonie, à destination des personnes lourdement handicapées, en suivant les exigences de Home-Based.

Ensuite, la suite du travail consiste à porter un regard critique sur l'application développée dans le cadre du stage. C'est-à-dire évaluer les forces et les améliorations possibles de l'application développée, autant du point de vue des fonctionnalités que du point de vue des interactions.

Avec l'expérience engrangée durant le stage, et afin d'améliorer l'application conçue durant celui-ci, l'étape suivante est de réaliser un état de l'art de ce qu'il existe dans la littérature en termes de modalités et de moyens d'interaction pour les personnes invalides.

Ensuite, le but est d'améliorer le système développé en fonction de ce qu'il existe afin de rencontrer le maximum d'exigences. Ces premières étapes doivent répondre à la première question de recherche.

Enfin, la dernière étape est d'établir une taxonomie des modalités d'interaction, afin de déterminer quelle modalité utiliser en fonction du handicap de la personne. Cette étape a pour objectif de répondre à la deuxième question de recherche.

## 1.4 Collaboration avec Jonathan Maes

L'immersion au sein de l'entreprise Home-Based a été effectué en lien avec Jonathan Maes. L'objectif du stage était de concevoir et développer deux applications VoIP pour le système Soline. Ces applications devaient principalement servir à passer des appels téléphoniques et à contrôler un portier de rue au travers d'un ordinateur. Les deux applications devaient répondre aux mêmes exigences, à la différence de la plateforme cible. D'un côté pour Windows dans le cas de ce travail, et de l'autre Android dans le cas de Jonathan Maes.

Pour certaines parties de la conception, une collaboration a été mise en place durant le stage.

Ce fut tout d'abord le cas pour la conception de l'architecture de l'interface graphique. En effet, le but était de proposer deux interfaces graphiques similaires, afin de faciliter l'utilisation de celles-ci simultanément.

De plus, le même procédé a été utilisé pour la conception des moyens d'interaction, toujours dans l'optique d'avoir un fonctionnement visuel similaire entre les deux applications.

Un autre cas de collaboration a été la représentation des données. Cela avait pour but d'avoir les mêmes informations pour les contacts ainsi que les historiques, et pour avoir la possibilité de s'échanger des données de contacts ou historiques d'appel.

Enfin, les rencontres avec les experts ont été effectuées ensemble.

Cette collaboration a eu lieu durant le stage. Par la suite, une séparation de nos objectifs a été effectuée, donnant lieu à deux travaux distincts. Ce présent travail se focalise sur les modalités et les moyens d'interaction pour les personnes lourdement handicapées, tandis que le travail de Jonathan a pour cadre les interfaces pour mobile.

## 1.5 Structure du travail

Le travail sera structuré comme suit : Tout d'abord, l'application développée durant le stage ainsi que les contacts avec les experts et l'utilisatrice seront expliquée au chapitre 2. Ensuite, le chapitre 3 portera sur le recul pris par rapport au stage. Le chapitre 4 présentera l'état de l'art, tandis que le chapitre 5 portera sur les améliorations apportées à l'application développée durant le stage. Le chapitre 6 présentera la conception d'une taxonomie des modalités d'interaction. Enfin, ce travail se terminera, au chapitre 7, par une conclusion et une présentation des travaux futurs.



## Chapitre 2

# Stage : une interface de téléphonie pour personnes handicapées

Ce chapitre porte sur la conception et le développement d'un prototype de l'application de téléphonie pour les personnes handicapées réalisée au sein de l'entreprise Home-Based. Ce chapitre est découpé en plusieurs parties, la première portant sur les objectifs de l'application, la deuxième sur la méthodologie appliquée pour la conception de l'application. La troisième partie porte sur la collecte des informations auprès d'experts du domaine. La quatrième partie présente la recherche de solutions existantes en termes d'applications VoIP et de bibliothèques graphiques. La cinquième partie détaille les méthodes VoIP, comme émettre un appel. Ensuite, la sixième partie porte sur la représentation et le traitement des données, notamment les contacts. La septième partie décrit les interactions avec l'application, alors que la huitième porte sur l'interface graphique. Enfin, la dernière partie présente le retour d'une utilisatrice de l'application.

### 2.1 Objectifs de l'application

Dans le cadre du stage, il était demandé de concevoir une application VoIP, qui répond à plusieurs exigences, définies au début du stage. Cette section détaille ces exigences.

La première est le fait d'avoir une interface graphique adaptée aux personnes à mobilité réduite. Elle doit donc être claire, épurée, intuitive et facile d'utilisation.

La deuxième exigence concerne les modalités d'interactions. L'interface doit être contrôlable par toutes les modalités d'interactions avec lesquelles Soline est compatible, ainsi que le couple clavier/souris. L'application doit également être capable de se connecter au système Soline, pour l'avertir notamment en cas d'appel entrant. Cela permettrait, par exemple, de couper le son de la télévision le temps de l'appel.

L'application doit avoir toutes les fonctionnalités d'une application VoIP, c'est-à-dire qu'elle doit pouvoir s'enregistrer sur un serveur utilisant le protocole SIP, savoir passer un appel audio et vidéo, recevoir un appel, jouer un code DTMF, etc.

Une quatrième exigence est d'avoir une gestion des contacts et des historiques d'appels. L'utilisateur doit pouvoir consulter ses contacts, les modifier, en ajouter ou encore en supprimer. Il doit également être capable de consulter et de supprimer ses historiques d'appels.



Enfin, l'application doit être capable de s'exécuter au minimum dans un environnement Windows. Cette exigence provient du fait que la reconnaissance vocale de Soline s'exécute sous Windows.

## 2.2 Méthodologie

Cette section détaille la méthodologie utilisée dans le cadre de l'immersion en stage chez Home-Based.

Premièrement a eu lieu une recherche de solutions existantes en matière de VoIP, ainsi qu'une prise en main de ces solutions. Le but étant d'avoir un prototype d'application VoIP fonctionnel, sans s'occuper de l'interface. Ce prototype devait également intégrer la gestion des contacts et des historiques d'appels.

Dans le même temps, une collecte d'informations auprès d'experts a eu lieu, afin d'avoir plus d'informations sur les personnes lourdement handicapées ainsi que sur les moyens et modalités d'interaction disponibles pour eux. Cela s'est traduit par une rencontre avec deux ergothérapeutes du CRETH<sup>1</sup> ainsi que plusieurs discussions avec les fondateurs de Home-Based.

Cette collecte a, par la suite, mené à une recherche sur l'interface graphique et les moyens d'interaction à mettre en place.

La quatrième étape est de développer un prototype d'application VoIP avec la prise en charge de l'interface graphique et des moyens d'interaction. La méthode de développement utilisée est une méthode agile. C'est-à-dire de petites phases de développement suivies de discussions chez Home-Based afin d'analyser la partie développée.

Enfin, une rencontre avec une future utilisatrice du système a eu lieu. Cette rencontre a permis de valider la partie interface et interaction du système développé dans le cadre du stage.

## 2.3 Collecte d'informations auprès d'experts

Cette section décrit la collecte des informations à propos des méthodes d'interaction existantes auprès d'experts. Le but de cette collecte était d'en apprendre d'avantage sur les modalités d'interaction existantes, sur les moyens d'interaction ainsi que sur les besoins des utilisateurs lourdement handicapés.

Au vu de la difficulté à trouver des utilisateurs invalides, il a fallu rapidement se tourner vers des experts en matière d'interactions pour ces utilisateurs.

Les premiers experts à avoir été consultés sont les fondateurs de Home-Based. Ils ont présenté les différentes modalités qu'ils proposent à leurs clients (section 1.2), la technique du balayage (section 4.2) ainsi que leur système Soline. Ils ont également été consultés tout au long de la conception des interactions et de l'interface graphique de l'application.

Les fondateurs de Home-Based ont également présentés l'application VoIP utilisée par leurs clients, Linphone. Cette dernière n'a pas été adaptée, et la voir à

---

1. <http://creth.be/>

permis de se rendre compte de l'importance de développer une application adaptée. Par exemple, pour ouvrir la porte lorsque quelqu'un sonne, il faut au minimum faire sept actions :

1. Appuyer sur le bouton **Décrocher**.
2. Bouger la fenêtre pour afficher la vidéo. En effet, la vidéo s'affiche dans une nouvelle fenêtre, positionnée derrière la fenêtre principale.
3. Cliquer dans la barre du menu sur **Option**.
4. Cliquer sur "Afficher le pavé numérique" (Voir figure 2.1).
5. Cliquer sur tous les numéros pour entrer le code pour déverrouiller la porte (Voir figure 2.2).
6. Fermer la fenêtre du pavé numérique.
7. Cliquer sur **Raccrocher**.

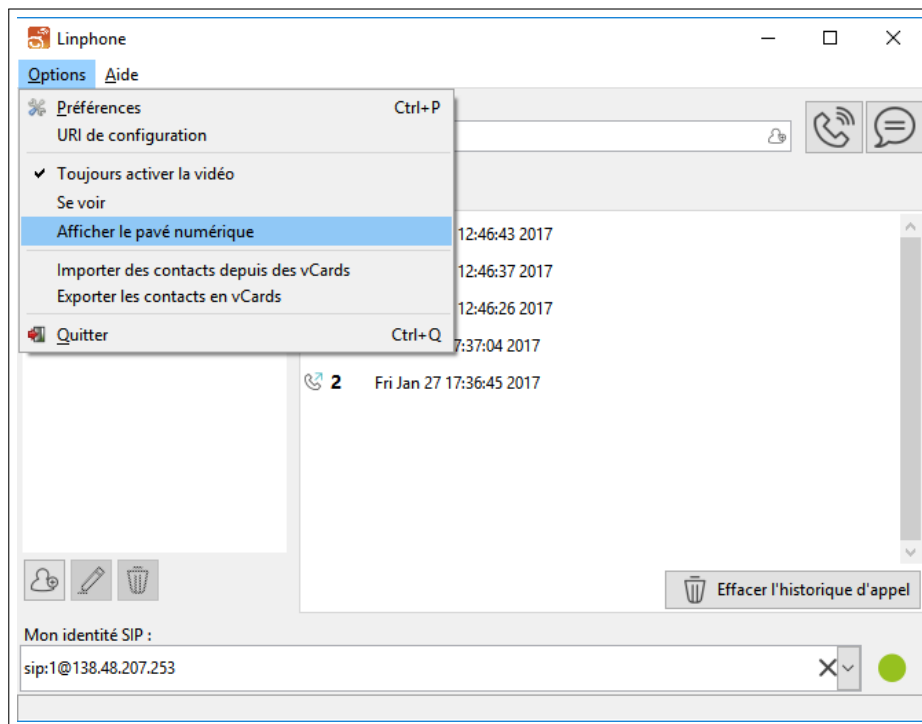


FIGURE 2.1 – Interface graphique de Linphone

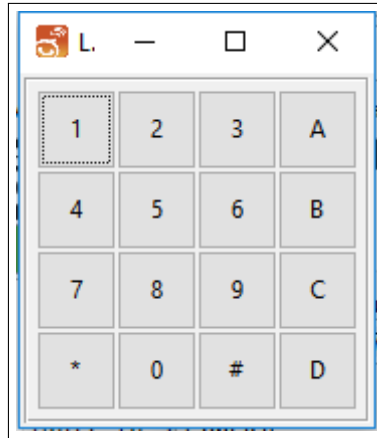


FIGURE 2.2 – Pavé numérique de Linphone

Donc, afin d’ouvrir la porte, il faut au minimum sept clics, nombre qui dépend de la longueur du code de la porte. Sans compter la complexité à appuyer sur des touches relativement petites, qui peut apporter son lot d’erreurs.

Les deuxièmes experts sont deux ergothérapeutes du CRETH, qui ont présenté d’autres modalités disponibles, comme les modalités utilisant les mouvements des yeux, et des exemples d’interfaces graphiques adaptées existantes. Ils ont également parlé des tags ARIA<sup>2</sup>. Ces tags sont utilisés dans le contexte du web afin de décrire au mieux une page web. Par exemple, il est possible de décrire le rôle d’un élément. Cela peut être un lien, un bouton, un choix multiple, etc. Ce sont des tags fort utiles notamment pour les personnes aveugles, car beaucoup de lecteurs d’écrans se basent sur ceux-ci pour lire la page.

Cette collecte d’information a contribué à la conception des interactions et de l’interface graphique, expliqués respectivement dans la section 2.7 et dans la section 2.8. Cela a également permis de se rendre compte des besoins et des difficultés que rencontrent au quotidien les personnes lourdement handicapées.

## 2.4 Solutions existantes

Avant de concevoir le prototype de l’application de téléphonie pour les personnes handicapées, il a fallu explorer plusieurs solutions existantes, dans le but de les adapter aux exigences demandées. Il a fallu également choisir une bibliothèque graphique pour le développement. Cette sous-section reprend toutes les solutions explorées ainsi que leurs avantages et inconvénients.

### 2.4.1 Solutions VoIP

#### Linphone<sup>3</sup>

La première solution explorée est Linphone. Linphone est un logiciel VoIP open-source développé par la société Belledonne Communications<sup>4</sup>. C’est le logiciel utilisé actuellement par Home-Based pour gérer la téléphonie.

2. <https://www.w3.org/TR/html-aria/>

3. <http://linphone.org/>

4. <http://www.belledonne-communications.com>

Les avantages de Linphone sont nombreux. Premièrement, Linphone est open-source. Cela signifie qu'il est modifiable, et qu'il est possible de changer notamment l'interface de l'application. Un deuxième avantage est que Linphone est multi-plateforme. Dans sa version *Desktop*, Linphone est compatible Windows, Linux et Mac OS.

Par contre, le gros point noir de Linphone est sa difficulté à être compilé dans un environnement Windows. En effet, le développement de Linphone se fait exclusivement dans un environnement Linux, et sa compilation n'est pas testée à chaque modification. De plus, le code fait référence à d'autres sous-modules (figure 2.3) qui composent Linphone, qui apportent également leur lot d'erreurs à la compilation. Pour preuve, le fichier contenant les instructions pour compiler sous Windows date du 24 février 2105, soit plus de deux ans. Entretemps, les sous-modules ont changé rendant totalement inutile ce fichier d'instruction.

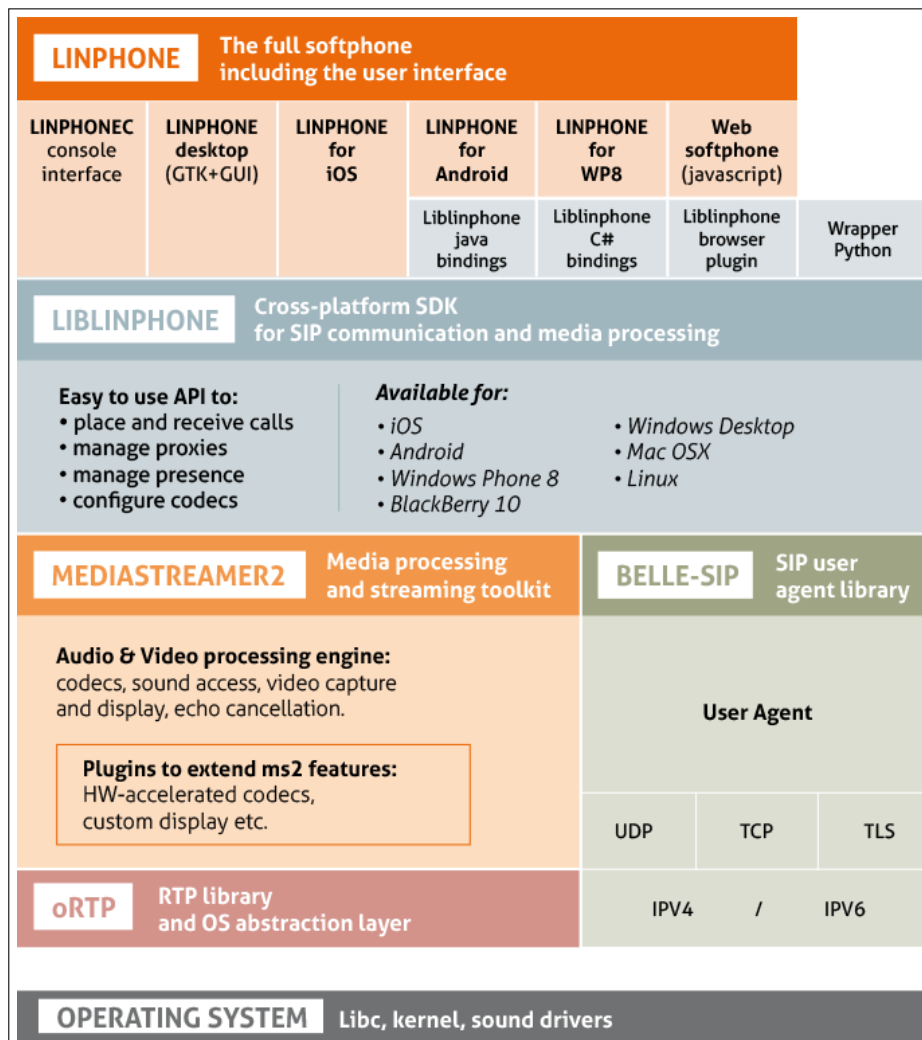


FIGURE 2.3 – Structure logicielle de Linphone<sup>5</sup>

Un autre inconvénient est son interface graphique, illustrée par la figure 2.1. Cette figure montre le cheminement qu'il faut faire pour atteindre le pavé numérique (figure 2.2), utile pendant un appel pour envoyer des codes DTMF. Un code DTMF

5. <http://www.linphone.org/technical-corner/linphone/overview>

est un code envoyé lorsqu'une touche du clavier d'un téléphone est pressée. L'utilité principale de ce code est de composer un numéro. Les codes DTMF sont également utilisés pour les services vocaux qui demandent d'appuyer une touche pour choisir une option (services clients, etc). L'inconvénient de l'interface est peu important dans ce cas-ci. En effet, dans la mesure où le projet est open-source, il est tout à fait possible de refaire complètement l'interface graphique.

## **Liblinphone**<sup>6</sup>

Liblinphone est une librairie créée par la société Belledonne Communications, c'est-à-dire la société qui développe Linphone. Ce dernier est en réalité basé sur Liblinphone, comme le montre la figure 2.3. Liblinphone fournit un kit de développement pour toutes les communications SIP ainsi que la gestion des médias (son et vidéo). SIP est un protocole utilisé pour gérer des sessions multimédias. Il est le plus utilisé dans les applications de téléphonies VoIP. Le gros point positif de Liblinphone est qu'il est fourni en version compilée. De plus, il dispose de toutes les fonctionnalités de Linphone, à l'exception de l'interface graphique.

## **Ozeki SDK**<sup>7</sup>

Ozeki SDK est un kit de développement pour applications VoIP basé sur SIP. Il est développé par la société hongroise Ozeki<sup>8</sup>. Ce kit est destiné à être utilisé en C#.

L'avantage de ce kit est sa documentation. Le kit est fourni avec beaucoup d'exemples d'une grande partie des fonctionnalités SIP (enregistrement sur un serveur SIP, appel, etc). De plus, le site contient également des tutoriels et le manuel d'utilisation du kit. Cela permet de développer une application de test assez rapidement.

Par contre, le kit dans sa totalité n'est pas disponible gratuitement. Il existe une version gratuite qui permet de tester quelques fonctionnalités SIP, mais pas toutes. Les fonctionnalités disponibles gratuitement ne sont pas suffisantes pour la réalisation de l'objectif du stage.

## **Choix**

Finalement, il a été décidé de choisir la solution de la bibliothèque Liblinphone, qui était la plus simple à mettre en œuvre. De plus, contrairement à Ozeki SDK, il était possible de développer la totalité du prototype en utilisant cette bibliothèque.

Liblinphone étant écrite pour être utilisée en C ou en C++, il a fallu trouver une bibliothèque graphique compatible avec ces langages.

---

6. <http://www.linphone.org/technical-corner/liblinphone/overview>

7. <http://www.voip-sip-sdk.com/>

8. <http://www.ozeki.hu/index.php>

## 2.4.2 Bibliothèques graphiques

Les deux principales bibliothèques pour le développement d'applications graphique en c++ sont GTK+<sup>9</sup> et Qt<sup>10</sup>.

### GTK+

GTK+ est une bibliothèque graphique multi-plateforme. Elle est également connue sous le nom de *GIMP Toolkit*. C'est la bibliothèque utilisée actuellement pour l'interface graphique de Linphone. Pour développer une application utilisant GTK+ dans un environnement Windows, il faut mettre en place tout un environnement de développement UNIX sous Windows, au travers du projet MYSYS2<sup>11</sup>.

### Qt

Qt est également une bibliothèque multi-plateforme. En plus de proposer tous les outils pour créer une interface graphique, elle propose son lot de fonctionnalités comme une gestion des XML, des sons, des bases de données, etc. Elle est fournie au travers d'un IDE (Qt Creator) qui contient toutes les librairies et toutes les dépendances. C'est donc une solution relativement simple à mettre en place.

### Choix

Le choix d'une bibliothèque graphique s'est porté sur Qt, principalement pour la simplicité à mettre en place une solution. De plus, beaucoup d'applications qui étaient développées avec GTK+ sont passées sous Qt, comme Wireshark<sup>12</sup>. Celles-ci indiquent que GTK+ ne couvre pas correctement le rendu sous toutes leurs plateformes (notamment Mac OS). De plus, l'utilisation de Qt est nettement plus simple, grâce notamment à la documentation très complète.

## 2.5 Méthodes VoIP

Cette section porte sur la conception et l'implémentation des méthodes VoIP. Il a fallu implémenter deux catégories de méthodes. La première concerne les méthodes d'appels, comme émettre un appel VoIP, en recevoir un ou encore envoyer un code DTMF durant un appel. La deuxième catégorie contient les méthodes de gestion des comptes, comme se connecter sur un serveur SIP ou s'en déconnecter.

Toutes ces fonctions doivent être liées au cœur de LibLinphone, un composant majeur de la bibliothèque du même nom. Ce dernier doit être unique tout au long de l'exécution du programme. Un tel comportement est modélisé par le *design pattern singleton*.

Le cœur de LibLinphone fonctionne via un système d'événements, liés à des méthodes à appeler lorsque cet événement se produit. Lors du démarrage du programme, il est nécessaire d'initialiser le cœur de LibLinphone. C'est à cette étape

---

9. <https://www.gtk.org/>

10. <https://www.qt.io/>

11. <http://www.msys2.org/>

12. <https://blog.wireshark.org/2013/10/switching-to-qt/>

qu'il faut indiquer quelles sont les méthodes qu'il faudra appeler lorsque le programme recevra un événement. Il est nécessaire de créer deux méthodes pour répondre aux différents événements : une portant sur les événements de compte et une portant sur les événements des appels. Ces deux méthodes vont être utiles pour la réception d'un appel, ainsi que la gestion de la connexion sur un serveur SIP.

Tout d'abord, la conception des deux méthodes liées aux événements seront abordées. Ensuite, les autres méthodes VoIP, comme émettre un appel ou se connecter sur un serveur SIP, seront décrites.

Les pseudos-codes présentés sont minimaux, c'est-à-dire exempts de logs ou de vérifications. Ils sont présents à titre d'information sur l'utilisation de Liblinphone.

## Événements de compte

Voici ci-dessous (figure 2.4) le pseudo-code de la méthode créée afin qu'elle soit appelée lorsque le programme reçoit un événement lié aux comptes SIP. Le programme peut recevoir ces trois événements importants :

- *LinphoneRegistrationOk* : Indique que la connexion au serveur SIP est réussie.
- *LinphoneRegistrationFailed* : Indique que la connexion au serveur SIP a échoué.
- *LinphoneRegistrationCleared* : Indique que la connexion au serveur SIP a été fermée.

```
1 static void registration_state_changed(LinphoneCore core,  
   ↪ LinphoneRegistrationState state){  
2     switch(state){  
3         case LinphoneRegistrationOk :  
4             // Connexion réussie  
5             break;  
6         case LinphoneRegistrationFailed :  
7             // Connexion échouée  
8             break;  
9         case LinphoneRegistrationCleared :  
10            // Déconnexion réussie  
11            break;  
12     }  
13 }
```

FIGURE 2.4 – Méthode "registration\_state\_changed"

Cette fonction offre par exemple la possibilité d'indiquer à l'utilisateur l'état actuel de sa connexion.

## Événements d'appels

La figure 2.5 représente, en pseudo-code, la fonction *call\_state\_changed*, méthode appelée lorsqu'un événement à propos d'un appel arrive. Voici les principaux événements que l'application peut recevoir à propos d'un appel :

- *LinphoneCallIncomingReceived* : Un appel entrant est détecté, le téléphone commence à sonner.
- *LinphoneCallOutgoingRinging* : Un appel sortant est en train de sonner de l'autre côté.
- *LinphoneCallConnected* : Un appel est connecté
- *LinphoneCallEnd* : Un appel est terminé

```

1  static void call_state_changed(LinphoneCore core, LinphoneCallState
   ↪ state){
2      switch(state){
3          case LinphoneCallIncomingReceived :
4              // Appel entrant détecté : lancement d'une sonnerie et
   ↪   affichage de la fenêtre d'appel entrant
5              break;
6          case LinphoneCallOutgoingRinging :
7              // Appel sortant en train de sonner
8              break;
9          case LinphoneCallConnected :
10             // Appel connecté : affichage de la fenêtre d'appel en
   ↪   cours
11             break;
12          case LinphoneCallEnd:
13             // Appel terminé : ajout d'une entrée dans le journal
   ↪   d'appel et retour au menu principal
14             break;
15     }
16 }

```

FIGURE 2.5 – Méthode "call\_state\_changed"

## Se connecter au serveur SIP

Pour se connecter sur un serveur SIP, il faut renseigner un nom d'utilisateur, un domaine, un mot de passe. Le nom d'utilisateur et le domaine forment l'adresse de l'utilisateur, sous la forme *nom\_utilisateur@domaine*.

La figure 2.6 représente, en pseudo-code, cette méthode.



```

1 static void login(Compte compte, LinphoneCore core){
2     // Création de l'adresse de l'utilisateur :
3     LinphoneAddress address = linphone_address_new(compte.user_name
4     ↪ + "@" + compte.domain);
5     // Création des informations d'authentification :
6     LinphoneAuthInfo auth_info =
7     ↪ linphone_auth_info_new(compte.user_name, compte.password);
8     // Fournir les informations d'authentification au cœur
9     ↪ LibLinphone. Ce dernier va se charger d'initialiser la
10    ↪ connexion
11    linphone_core_add_auth_info(core, auth_info);
12 }

```

FIGURE 2.6 – Méthode "login"

## Se déconnecter du serveur SIP

Pour se déconnecter d'un serveur SIP, il est nécessaire de supprimer la configuration de l'utilisateur. Lors de l'inscription, une configuration est créée pour l'utilisateur.

La figure 2.7 représente, en pseudo-code, cette méthode.

```

1 static void login(Compte compte, LinphoneCore core){
2     // Création de l'adresse de l'utilisateur :
3     LinphoneAddress address = linphone_address_new(compte.user_name
4     ↪ + "@" + compte.domain);
5     // Création des informations d'authentification :
6     LinphoneAuthInfo auth_info =
7     ↪ linphone_auth_info_new(compte.user_name, compte.password);
8     // Fournir les informations d'authentification au cœur
9     ↪ Linphone. Ce dernier va se charger d'initialiser la
10    ↪ connexion
11    linphone_core_add_auth_info(core, auth_info);
12 }

```

FIGURE 2.7 – Méthode "login"

## Initier un appel

La méthode pour émettre un appel, il consiste simplement à donner une adresse SIP (*nom\_utilisateur@domaine*) au cœur de Linphone. Cette méthode est représentée par la figure 2.8

```

1 static void call(String user, String domain LinphoneCore core){
2     // Création de l'adresse SIP
3     String address = user + "@" + domain;
4     // Transmission au cœur de Linphone
5     linphone_core_invite(core, address);
6 }

```

FIGURE 2.8 – Méthode "call"

## Envoyer un code DTMF

De nouveau, la fonctionnalité d'envoi d'un DTMF est assez simple, grâce à Linphone. Il suffit simplement de transmettre au cœur de Linphone le code à envoyer. La figure 2.9 représente cette méthode.

```

1 static void send_DTMF(LinphoneCall call, String dtmf){
2     // Transmission du code DTMF à l'appel
3     linphone_call_send_dtmfs(call, dtmf);
4 }

```

FIGURE 2.9 – Méthode "send\_DTMF"

## 2.6 Données

Cette section décrit les choix de conception qui ont été posés pour la gestion des données. Les données à représenter sont les comptes SIP, les contacts et les historiques d'appel. Les points de vue de la représentation des données, de leur stockage ainsi que de leurs traitements (sauvegarde, chargement, utilisation) seront abordés.

### Représentation

La figure 2.10 illustre le schéma relationnel des données de l'application.

Un compte représente les informations de connexion d'un compte SIP. Il faut donc un nom d'utilisateur, un domaine ainsi qu'un mot de passe. L'adresse SIP du compte est formée par le nom d'utilisateur et le domaine, sous la forme de *nom\_utilisateur@domaine*.

Un historique représente une entrée du journal d'appel, afin de garder trace de tous les appels. Il faut donc la date de l'appel, le numéro de l'interlocuteur ainsi que la durée de l'appel. En plus de ces informations, il faut pouvoir distinguer si l'appel est entrant ou sortant. Il faut également savoir identifier si l'utilisateur a décroché ou non l'appel. Cette dernière information est représentée par le champ *manque*. S'il est à *true*, cela signifie que l'utilisateur n'a pas décroché l'appel, et donc l'appel est manqué. Le numéro peut être relié à un contact, dans le cas où il est connu.

Un contact doit avoir un nom et un prénom. Dans le cadre de ce travail, il existe deux types de contact : en effet, en plus des contacts ayant une adresse SIP (nom\_utilisateur@domaine), nous devons gérer les contacts ayant un numéro de téléphone classique (0000/00.00.00). De là, deux solutions apparaissent pour les contacts ayant les deux numéros : soit créer deux champs dans le contact, un pour l'adresse SIP et un pour le numéro classique, ou créer deux contacts différents, un avec l'adresse SIP, et l'autre avec le numéro classique. En collaboration avec Jonathan Maes, le choix a été fait de maximiser la simplicité pour les utilisateurs. Il a donc été décidé de séparer le carnet d'adresses en deux carnets d'adresses différents. Et par conséquent de créer deux contacts différents pour une personne ayant les deux numéros. La même question s'est posée pour les numéros de téléphone fixe et mobile. Le même choix a été posé, c'est-à-dire les séparer en deux contacts, avec par exemple *Alice Fixe* et *Alice GSM* comme noms de contact.

En plus de ces attributs, un contact possède un surnom, qui est surtout utile pour la reconnaissance vocale de Soline. Cela permet aux utilisateurs d'appeler plus simplement leurs contacts, que de devoir dire le nom complet.

Enfin, un dernier attribut est la sonnerie. La sonnerie est un champ texte, qui représente un chemin relatif qui pointe vers un fichier de son.

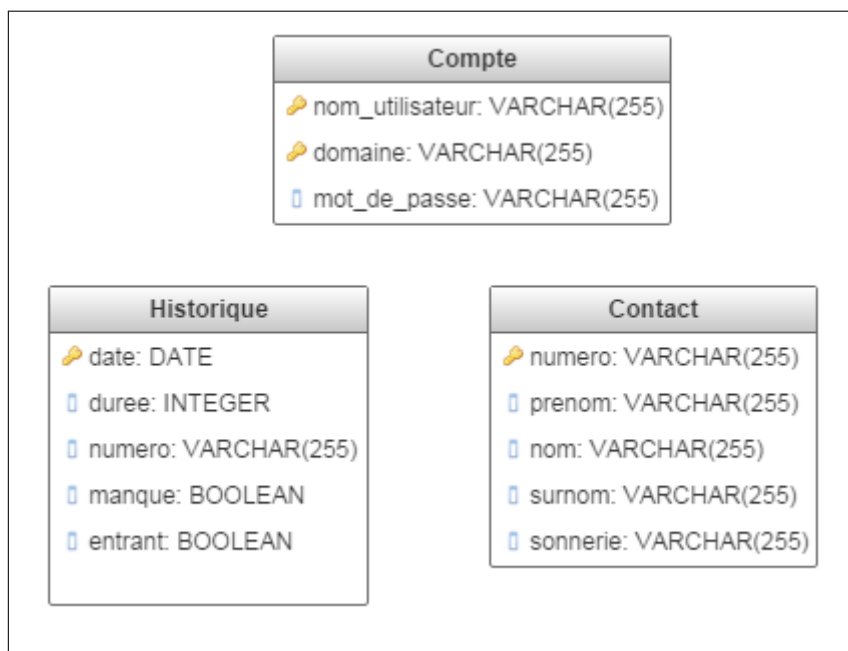


FIGURE 2.10 – Représentation des données

## Stockage & Traitement

Les données devant être stockées de façon permanente, le choix a été fait d'utiliser une base de données SQLite.

Ces données sont extraites de la base de données au démarrage de l'application. Chaque fois que l'utilisateur ajoute un contact, il est automatiquement ajouté dans la base de données. De même pour la suppression ou la modification, qui sont répliquées directement dans la base de données. Comparé à une politique de sauvegarde des données à la fermeture du logiciel, cette solution permet, en cas de défaillance de

l'application, de ne pas avoir d'impact sur les données des contacts. De même pour les historiques, qui sont ajoutés dès qu'un appel est terminé.

Ces différents choix seront discutés dans le chapitre 3.

## 2.7 Interaction

Cette section décrit comment sont gérées les différentes modalités d'interaction ainsi que les différents moyens d'interaction.

L'objectif de l'application, en terme d'interaction, est d'être contrôlable par le plus possible de modalités d'interaction, comme les boutons ou la voix. Dans le cadre du stage, il était demandé de se focaliser sur les boutons poussoirs, les télécommandes ainsi que la voix.

Le point commun entre toutes les modalités, hormis la voix, est que, quel que soit le dispositif utilisé, ce dernier simule un appui sur une touche du clavier. Il est donc assez aisé de prendre en charge ces différentes modalités.

La technique d'interaction mise en place est le balayage (voir section 4.2). Sans entrer dans les détails, le balayage consiste à parcourir les éléments de l'écran séquentiellement, et ensuite sélectionner un élément pour l'activer. En général, l'élément actif est mis en surbrillance. Le balayage se matérialise ici par un cadre rouge en pointillé autour de l'élément actuellement actif.

Le balayage peut fonctionner de plusieurs manières en fonction des dispositifs que l'utilisateur possède. S'il ne possède qu'un seul bouton, le balayage changera automatiquement l'élément actif à intervalle régulier, toutes les secondes par exemple. Le bouton de l'utilisateur lui servira alors à sélectionner l'élément actif. Dans le cas où il a deux boutons, il peut faire avancer lui-même le curseur. Il peut également y en avoir cinq, et dans ce cas faire bouger l'élément actif dans les quatre directions (haut, bas, gauche et droite).

Afin de gérer cela dans l'application, il a fallu concevoir un protocole qui permet de prévenir l'application du nombre de boutons que possède l'utilisateur. Ce protocole a été conçu avec l'aide de Home-Based ainsi que celle de Jonathan Maes.

Le protocole définit huit actions afin de contrôler le balayage sur une interface. Ces huit actions sont reprises dans le tableau 2.1.

Ok	Sélection de l'élément actif
Back	Retour à l'écran précédent
Next	Avance à l'élément suivant
Prev	Retour à l'élément précédent
Up	Accès à l'élément positionné au dessus
Down	Accès à l'élément positionné en dessous
Left	Accès à l'élément positionné à gauche
Right	Accès à l'élément positionné à droite

TABLE 2.1 – Actions disponibles pour contrôler une interface

De ces actions, il est possible de déduire dix modes d'utilisations de l'application en fonction du nombre de dispositifs que possède l'utilisateur. Ces dix modes sont repris dans le tableau 2.2.

Modes	Nombre de boutons	Actions
1	1	Ok
2	2	Ok - Back
3	2	Ok - Prev
4	2	Ok - Next
5	3	Ok - Prev - Back
6	3	Ok - Next - Back
7	3	Ok - Next - Prev
8	4	Ok - Next - Prev - Back
9	5	Ok - Up - Down - Left - Right
10	6	Ok - Up - Down - Left - Right - Back

TABLE 2.2 – Modes d'utilisations d'une application

L'interface graphique est capable de s'adapter en fonction du mode d'utilisation actif. Un exemple possible est l'ajout d'un bouton **Retour** lorsque le mode d'utilisation ne prévoit pas d'action **Back** (1, 3, 4, 7, 9). De même, le moyen d'interaction est capable de s'adapter. C'est le cas lorsque l'utilisateur n'a qu'un bouton (Mode 1), le parcours séquentiel des éléments démarre, et change d'élément toutes les deux secondes (le temps est paramétrable).

Une application peut être contrôlée en même temps par plusieurs dispositifs. Dans ce cas, le protocole prévoit de placer l'interface de l'application concernée dans le mode le plus contraignant. Par exemple, un utilisateur possède un unique bouton poussoir près de son lit (Mode 1), et une télécommande un peu plus loin (Mode 10). S'il choisit de connecter les deux dispositifs à l'application, cette dernière se met en mode "1", qui est le plus contraignant. En effet, s'il est sur son lit et que sa télécommande n'est pas à portée, il doit être capable de contrôler son application.

Le protocole prévoyait également le contrôle par la voix. En utilisant un dispositif de reconnaissance de paroles, il est possible d'envoyer à ce dispositif la liste des commandes vocales que l'application gère. Lorsque le dispositif de reconnaissance vocale détecte une commande gérée par l'application, il l'envoie à cette dernière. L'application développée ici gère l'envoi de la liste des commandes et la réception d'une commande, mais n'a pas été testée.

Du point de vue technique, le protocole requiert un élément central, qui gère les dispositifs disponibles pour l'application. C'est à cet élément central qu'il revient de décider du mode d'utilisation à utiliser et de transmettre les actions produites par les dispositifs. Par exemple, un appui sur un bouton pourra transmettre à l'application la commande **Ok**, si c'est cette commande qui est associé à ce bouton.

En plus du balayage, il a fallu trouver une solution pour le parcours des listes de contacts. En effet, si l'utilisateur dispose d'une centaine de contact, arriver au cinquantième contact va être très long. La solution mise en place consiste à placer deux listes à l'écran : une contenant les lettres de l'alphabet et une contenant les contacts, triés par ordre alphabétique. La première liste ne contient pas toutes les

lettres de l'alphabet, mais uniquement celles qui apparaissent à la première lettre d'un contact. Voici un exemple pour illustrer cette solution : Prenons une liste de sept noms (Alice, Bastien, Bertrand, Bob, Coralie, George et Trudy). La liste des lettres contiendra le "A", le "B", le "C", le "G" et le "T", comme illustré à la figure 2.11. Au début de la recherche, le champ de recherche est vide.

Champ de recherche : ____	— A	— Alice
	— B	— Bastien
	— C	— Bertrand
	— G	— Bob
	— T	— Coralie
		— George
		— Trudy

FIGURE 2.11 – Solution liste des contacts : champ de recherche vide

Prenons comme exemple que l'utilisateur veuille sélectionner Bob. Il va sélectionner dans la liste de gauche la première lettre de Bob, donc "B". Le "B" va venir s'ajouter au champ de recherche. Les deux listes vont s'adapter. Celle de droite va conserver uniquement les prénoms qui commencent pas le champ de recherche, c'est-à-dire "B". Celle de gauche va contenir les lettres suivantes des contacts encore affichés, c'est-à-dire "A", "E" et "O". De plus, il va contenir un nouveau caractère, le moins (-), qui permet de retirer une lettre au champ de recherche. Ceci est illustré à la figure 2.12.

Champ de recherche : B____	— A	— Bastien
	— E	— Bertrand
	— O	— Bob
	— -	

FIGURE 2.12 – Solution liste des contacts : champ de recherche = "B"

Il reste donc trois contacts dans la liste. Pour sélectionner "Bob", il faut sélectionner la lettre "O". La figure 2.13 représente la situation avec le champ de recherche mis à "BO". Il ne reste plus qu'une entrée dans les contacts, l'utilisateur peut donc passer dans la liste de droite et sélectionner son contact. Il est à noter que l'utilisateur peut à tout moment se rendre dans la liste de droite et naviguer dans cette liste là.

Champ de recherche : BO____	— B	— Bob
	— -	

FIGURE 2.13 – Solution liste des contacts : champ de recherche = "BO"

Du point de vue des modalités de sorties, l'application gère le *Text-To-Speech*, c'est-à-dire la lecture des éléments à l'écran. L'outil de lecture des éléments fourni sur Windows ne fonctionnait pas très bien avec le système, il a donc fallu ré-implémenter les fonctionnalités de lecture des éléments.

## 2.8 Interface

Cette section décrit la conception ainsi que l'implémentation de l'interface graphique. Celle-ci a été conçue sur base des conseils de Home-Based et du CRETH (section 2.3).

Pour les personnes invalides, l'interface se doit d'être la plus simple possible. Il est indispensable de se préoccuper des fonctionnalités importantes de l'application. Dans notre cas, cela concerne les appels, les contacts et les historiques d'appels. En effet, il est inutile pour la personne invalide de configurer son compte SIP, de renseigner les périphériques d'entrées et sorties pour la vidéo et le son, de configurer les bons codecs à utiliser, etc. Toute cette partie configuration est à faire lors de l'installation de l'application, par une personne valide.

L'interface à concevoir doit donc remplir quatre grosses fonctionnalités : la gestion des appels, des contacts et des historiques, ainsi que la fermeture de l'application.

Il est nécessaire que les éléments de l'interface prennent le plus de place possible et soient lisibles. Pour cette raison, l'interface est affichée en plein écran. De plus, afin de garantir une certaine cohérence, cette interface est similaire à celle conçue par Jonathan Maes, qui a développé une application VoIP pour mobile, chez Home-Based également.

La structure générale d'une page contient quatre cases, à définir avec les actions possibles pour chaque page. Les interfaces de ces différentes pages seront détaillées par la suite dans cette section. Chaque case contient le titre de l'action, ainsi qu'une image. A noter que, d'après les conseils de Home-Based, la police utilisée pour l'écriture est la police "Optima", qui permet une très bonne lisibilité. La figure 2.14 illustre le menu principal de l'application, où la structure de quatre cases est présente.

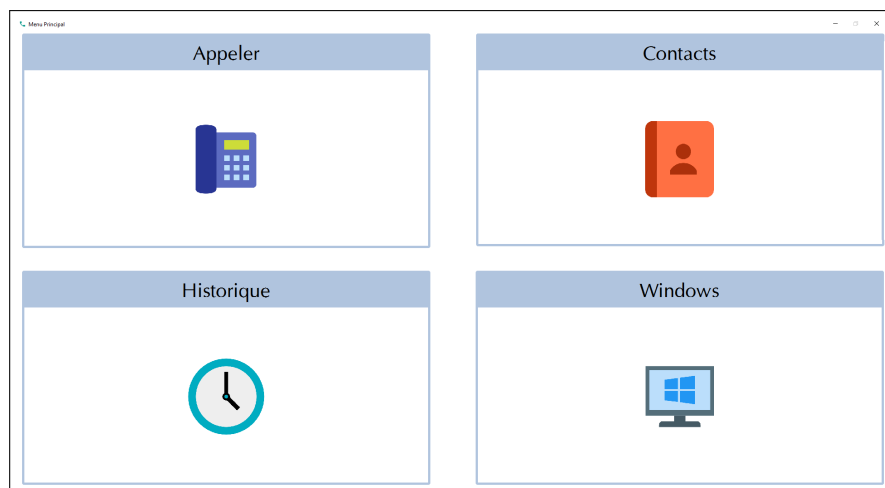


FIGURE 2.14 – Menu principal

Lorsqu'il s'agit de présenter de l'information, comme les listes de contacts ou encore les détails des contacts, les boutons sont alignés sur la gauche, afin de maximiser l'espace disponible pour afficher l'information.

Les sections ci-dessous décrivent l'ensemble des pages de l'interface. Comme expliqué à la section 2.7, le cadre en pointillé autour d'une case représente l'élément actif du balayage.

## 2.8.1 Gestion des appels

La gestion des appels concerne la composition d'un numéro, la période entre la fin de la composition du numéro et le début de l'appel (sonnerie chez l'interlocuteur), la réception d'un appel, l'appel en cours, l'affichage du pavé numérique pendant un appel (afin d'envoyer les codes DTMF) et enfin l'écran de fermeture d'un appel.

### Composition d'un numéro

La page de composition d'un numéro sert à introduire un numéro afin d'appeler ce dernier. Il contient donc tous les chiffres, ainsi que les caractères spéciaux utilisés sur les téléphones (\*, # et +). De plus, il contient également trois boutons, **Appeler**, **Retour**, pour retourner à l'écran précédent, et **Effacer** pour effacer un caractère du champ des caractères entrés.

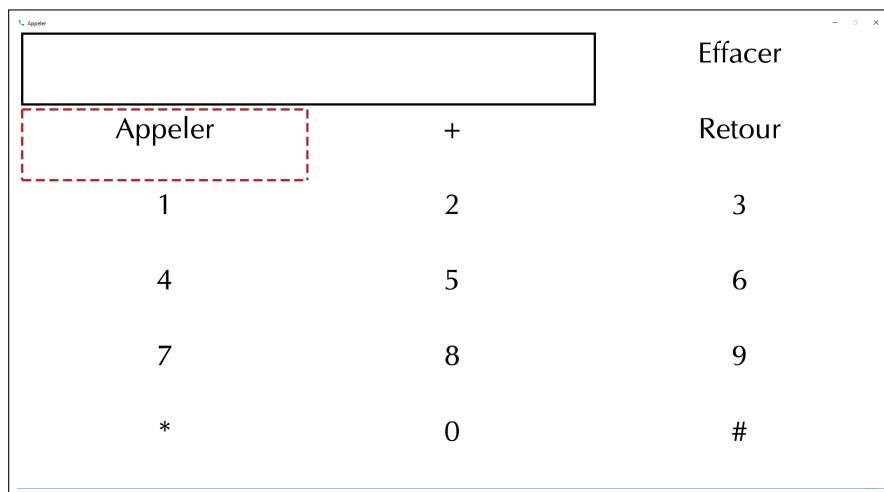


FIGURE 2.15 – Composition d'un numéro

### Numérotation en cours

Lorsque l'utilisateur décide d'appeler un contact ou bien un numéro qu'il a composé à l'aide de la page précédente, la sonnerie de l'interlocuteur commence. Sur la figure 2.16, l'utilisateur est en train d'appeler la cuisine. Il a encore la possibilité de raccrocher.



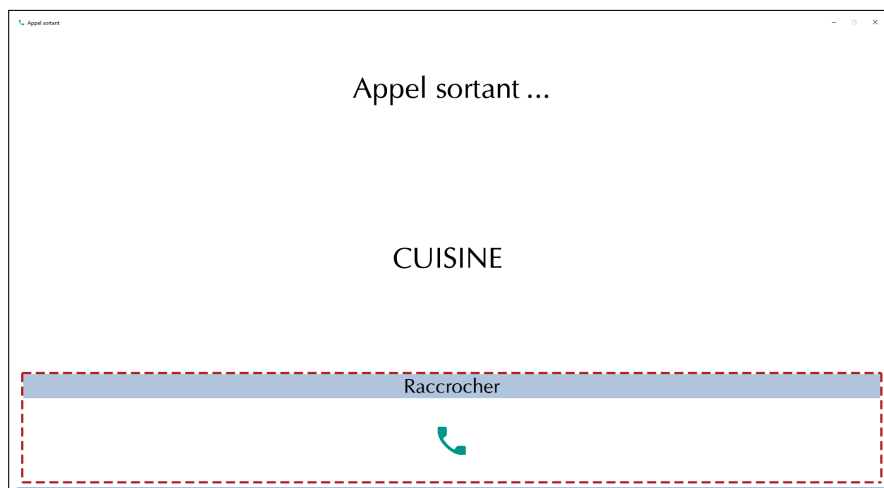


FIGURE 2.16 – Numérotation en cours

### Réception un appel

Cette page s’affiche lorsque l’utilisateur reçoit un appel. Le numéro de l’interlocuteur est affiché à l’écran. Si le numéro correspond à un contact, c’est le nom de ce dernier qui s’affiche. L’utilisateur a deux actions possibles, soit décrocher l’appel, soit mettre la sonnerie en silence (volume à 0).



FIGURE 2.17 – Réception d’un appel

### Appel en cours

L’interface qui s’affiche lorsqu’un appel est en cours est celle-ci. Elle permet à l’utilisateur d’ouvrir la pavé numérique afin d’envoyer un code DTMF (**Pavé**), de baisser ou augmenter le volume de l’appel (**Volume -**, **Volume +**) et de raccrocher (**Raccrocher**). De plus, si l’appel est en provenance d’une adresse SIP enregistrée comme étant une porte d’entrée, l’option **Ouvrir la porte** s’affiche afin d’envoyer à la porte les codes DTMF correspondants au code d’ouverture de la porte. Cela permet d’éviter à l’utilisateur de devoir entrer lui-même les codes DTMF pour la porte.

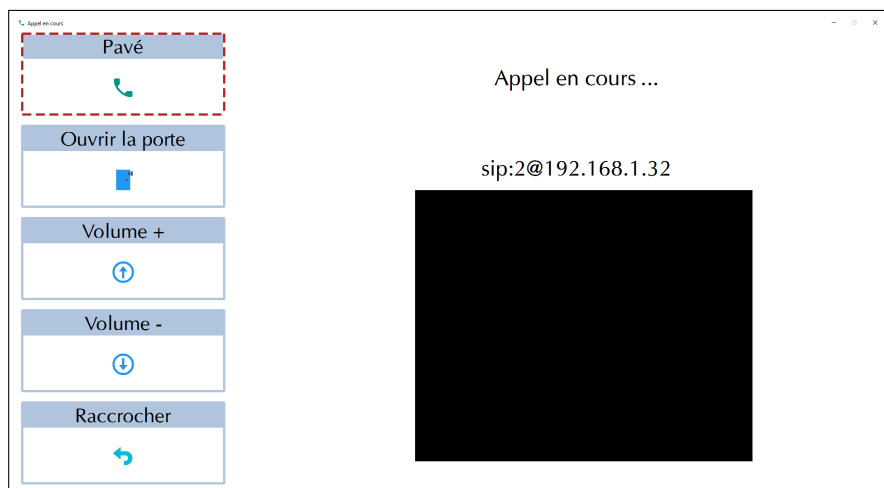


FIGURE 2.18 – Appel en cours

### Pavé numérique

Le pavé numérique est la page qui s’affiche lorsque, lors d’un appel en cours, l’utilisateur appuie sur le bouton **Pavé**. Cela affiche le pavé numérique avec les dix chiffres ainsi que les caractères spéciaux utilisés pour les codes DTMF (\* et #). Dès que l’utilisateur appuie sur un caractère, cela envoie directement le code DTMF correspondant.

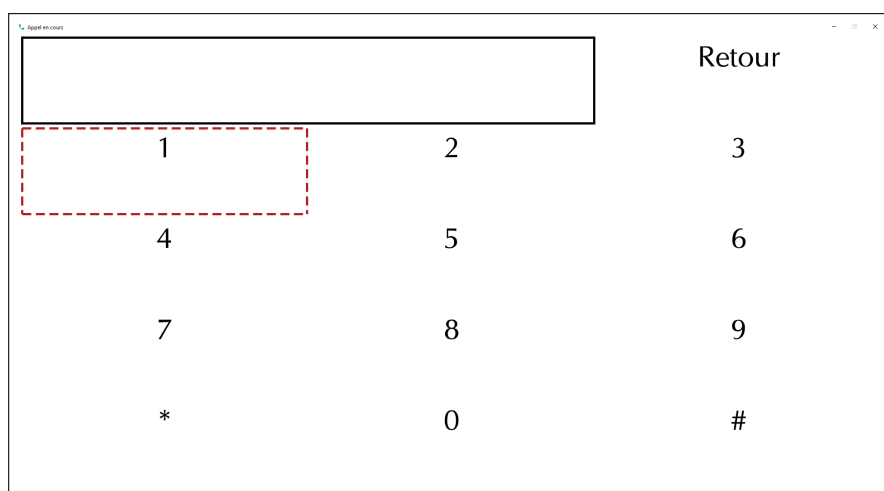


FIGURE 2.19 – Pavé numérique

## 2.8.2 Gestion des contacts

La gestion des contacts concerne le menu des contacts, l’affichage de tous les contacts ou d’un seul et enfin l’ajout, la modification et la suppression d’un contact.

### Menu des contacts

Les contacts sont séparés en deux groupes (section 2.6), nommés les contacts du téléphone et les contacts internes. Ces derniers contiennent les contacts ayant une adresse SIP. Dans la pratique, de tels contacts seront en réalité la porte d’entrée, un

poste dans la cuisine, etc. C'est la raison pour laquelle leur catégorie s'appelle les contacts internes.

Sur la page, apparaissent donc deux actions, qui servent à aller consulter les contacts d'une catégorie. En plus de ces deux actions, une troisième consiste à ajouter un contact. Enfin, la dernière est utilisée afin de retourner à l'écran précédent (**Menu principal**). A noter que le bouton indique à l'utilisateur sur quel écran il va retourner, et lui permet donc de se représenter correctement l'arborescence de l'interface.

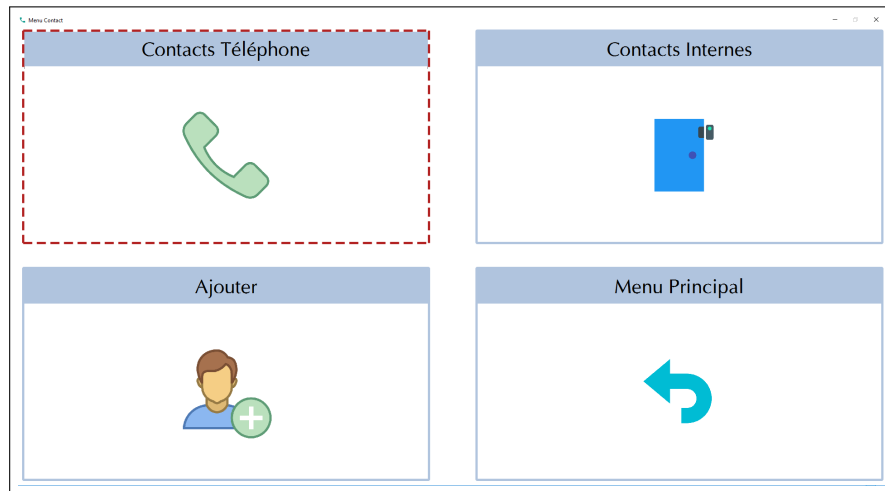


FIGURE 2.20 – Menu des contacts

### Affichage de tous les contacts

La page présentée ici est la page affichée lorsque l'utilisateur demande à consulter les contacts. L'interface est la même, que ce soit pour les contacts du téléphone que pour les contacts internes. Les boutons sont ici alignés sur le côté gauche, pour laisser le maximum de place à l'information à afficher. Comme expliqué dans la section interaction (section 2.7), il y a deux listes, une pour les lettres et une pour les noms de contacts. Les boutons **Monter** et **Descendre** servent à naviguer dans la liste actuellement sélectionnée. Par défaut, c'est la liste des contacts qui est sélectionnée. Afin de changer de liste, il faut actionner le bouton **Recherche**. Le bouton **Sélectionner** sert à activer l'élément actif de la liste (représenté par le cadre bleu). En actionnant une lettre, le filtrage s'effectue. En appuyant sur un nom de contact, l'utilisateur arrive sur la page d'affichage du contact. Enfin, un dernier bouton est présent afin de retourner à l'écran précédent (**Menu contact**).

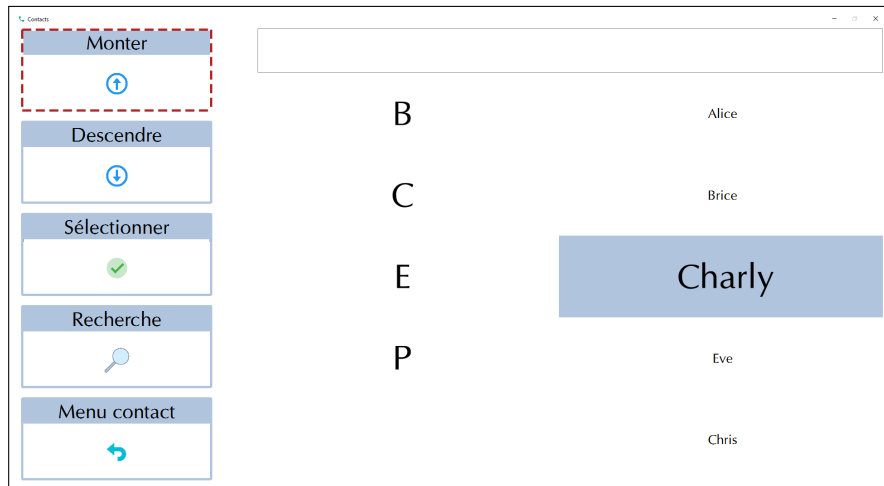


FIGURE 2.21 – Liste des contacts

### Affichage d'un contact

L'affichage d'un contact reprend également la structure d'une page informative (boutons sur la gauche). Sur cette page, nous pouvons retrouver les informations du contact (**Prénom**, **Nom**, **Numéro** et **Surnom**), ainsi que quatre actions disponibles. Ces quatre actions servent à appeler le contact (**Appeler**), à le modifier (**Modifier**) à le supprimer (**Supprimer**) ainsi qu'une action pour retourner à l'écran précédent (**Retour**).

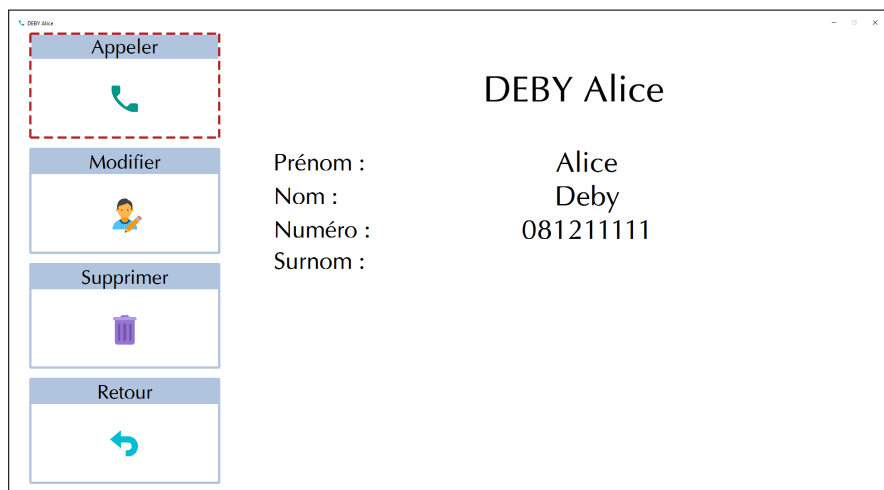


FIGURE 2.22 – Affichage d'un contact

### Ajout et modification d'un contact

La page d'ajout et de modification d'un contact est la même, à la seule différence que les champs **Prénom**, **Nom**, **Surnom** et **Numéro** sont pré-remplis dans le cas de la modification d'un contact. Ici, seuls deux boutons sont nécessaires : un pour confirmer l'ajout ou les modifications (**Confirmer**) et un pour annuler l'ajout ou les modifications et retourner à l'écran précédent (**Menu principal**).

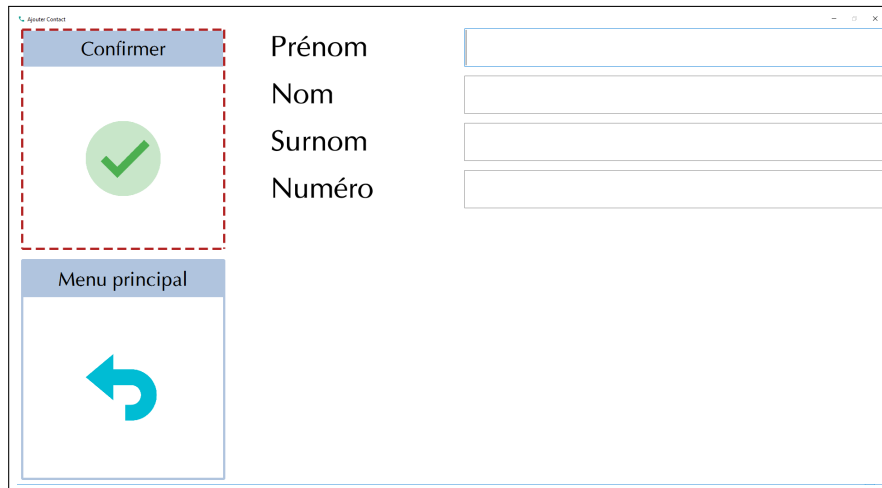


FIGURE 2.23 – Ajout d’un contact

### 2.8.3 Gestion des historiques d’appels

La gestion des historiques d’appels concerne le menu des historiques d’appels, l’affichage de tous les historiques d’appels ou d’un seul et enfin la suppression des historiques d’appels.

#### Menu des historiques

Tout comme les contacts, l’application distingue l’historique d’appels des contacts du téléphone et des contacts internes. Nous retrouvons donc une structure similaire à la page du menu des contacts. La différence est que l’action d’ajout d’un contact est remplacée par la suppression des historiques d’appels.

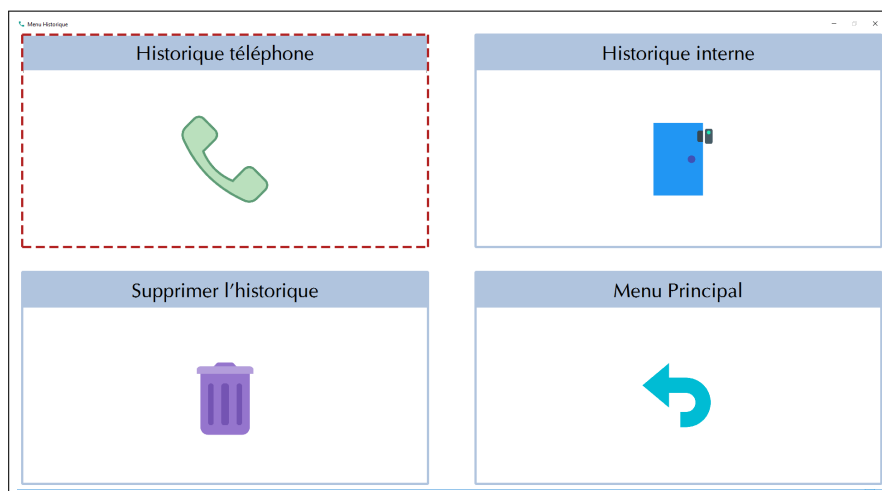


FIGURE 2.24 – Menu des historiques d’appels

#### Affichage de tous les historique d’appels

Contrairement aux contacts, il n’est pas nécessaire de rechercher dans les historiques d’appels. Il n’y a donc, sur cette page, qu’une seule liste, qui reprend tous les historiques d’appels par date. Il ne suffit donc que des boutons **Monter**, **Descendre**

et **Sélectionner** pour naviguer dans la liste. De nouveau, un quatrième bouton permet de retourner à l'écran précédent (**Menu historique**). L'interface est la même pour les historiques d'appels du téléphone et les historiques d'appels internes. La seule différence est dans le contenu de la liste affichée.

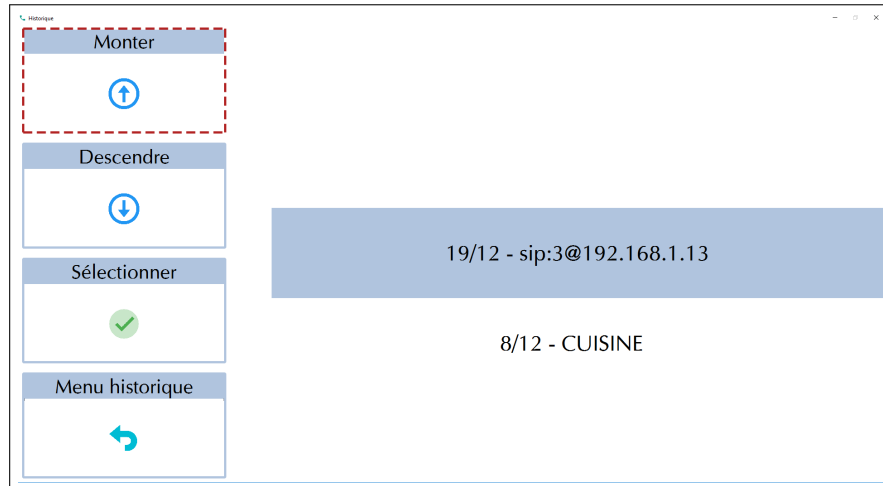


FIGURE 2.25 – Affichage des historiques d'appels

### Affichage d'un historique d'appels

Cette interface est atteinte lorsqu'un historique d'appels est sélectionné. On y retrouve les informations sur le type d'appel (**Appel entrant**, **Appel sortant** et **Appel manqué**), la date et l'heure de l'appel ainsi que sa durée. Il est possible d'appeler le contact ou le numéro lié à l'historique d'appel (**Rappeler**), de modifier le contact associé à ce numéro, s'il existe, ou d'ajouter un contact lié à ce numéro dans le cas inverse. Il est également possible de supprimer l'entrée de l'historique d'appels (**Supprimer**) et de retourner à l'écran précédent (**Retour**).

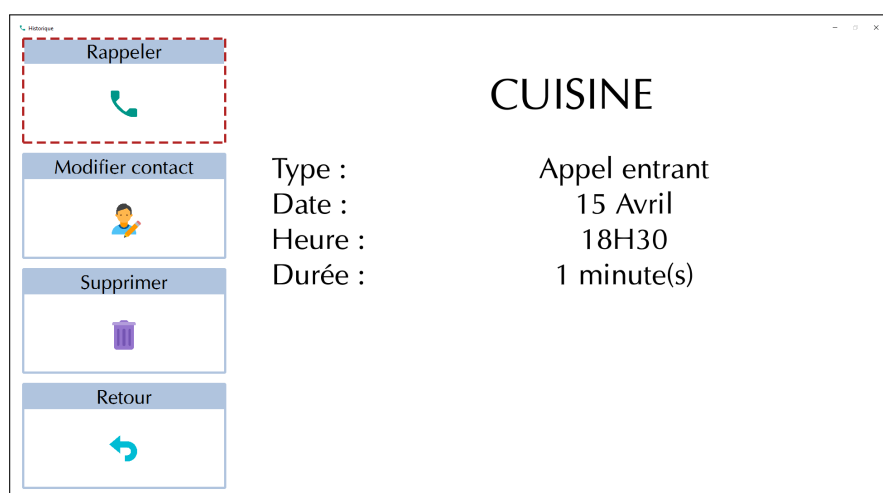


FIGURE 2.26 – Affichage d'un historique d'appel

### 2.8.4 Fermeture de l'application

La fermeture de l'application correspond à placer l'application au second plan. Elle tourne toujours en tâche de fond, notamment en attendant un appel, mais ne s'affiche plus sur tout l'écran. Néanmoins, une petite fenêtre se place sur un endroit de l'écran, afin de pouvoir revenir aisément à l'application (figure 2.27).

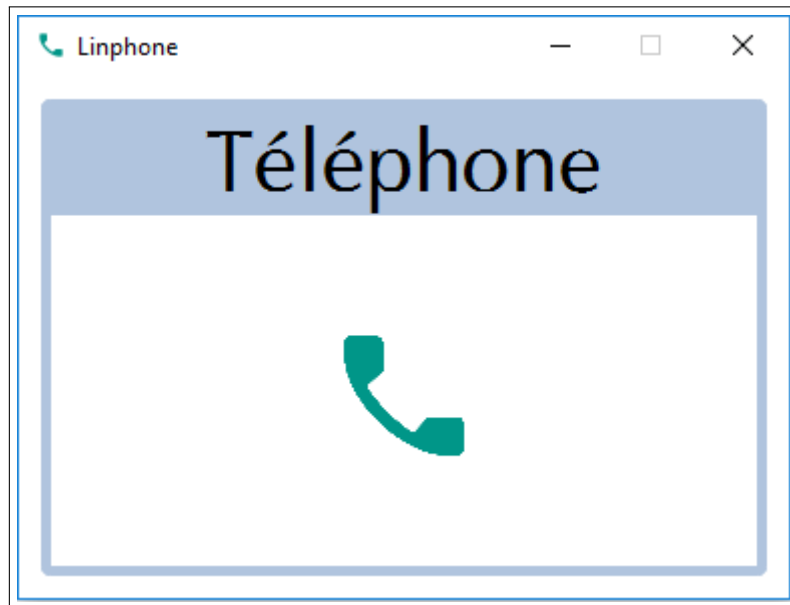


FIGURE 2.27 – Fenêtre placée sur un endroit de l'écran

### 2.8.5 Boîtes de dialogue

Les boîtes de dialogue contiennent les messages d'erreurs d'interrogations et d'informations. Ils ont tous la même structure. Le message sur le dessus de l'interface et les boutons en dessous. Pour le message d'information et le message d'erreur, un seul bouton est nécessaire : **Ok**. Pour le message d'interrogation, il faut deux boutons, **Oui** et **Non**. Les sous-sections suivantes présentent chaque type de boîtes de dialogue.

#### Appel terminé

Lorsque l'appel est terminé, un message d'information est affiché à l'écran.

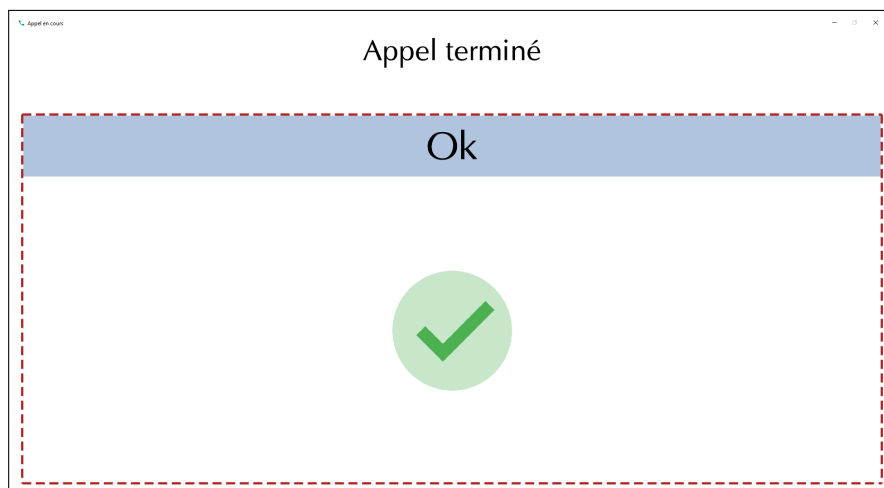


FIGURE 2.28 – Appel terminé

### Compte SIP non connecté

Lorsque le compte SIP de l'application n'est pas connecté au serveur SIP, l'application affiche ce message d'erreur chaque fois que l'utilisateur souhaite passer un appel. Ce message est également affiché si le compte se déconnecte pendant l'utilisation de l'application.

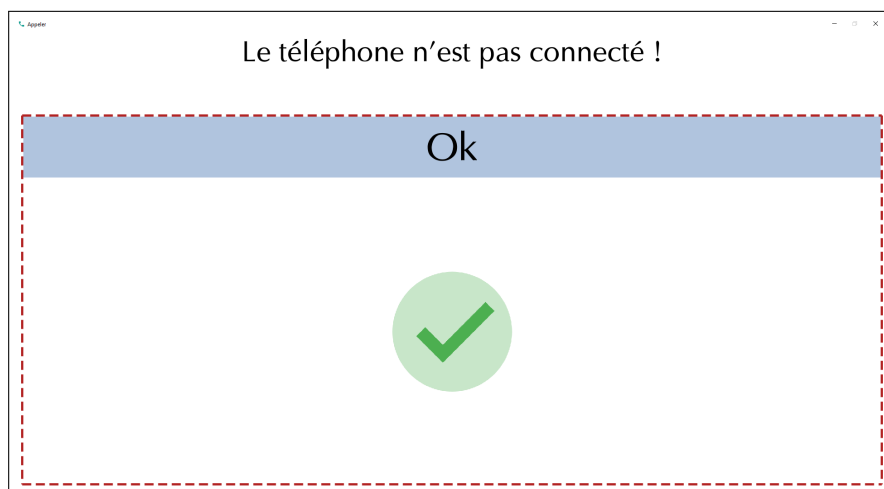


FIGURE 2.29 – Compte SIP non connecté

### Suppression des historiques d'appel

Lorsque l'utilisateur désire supprimer l'ensemble des historiques d'appel, l'utilisateur doit confirmer son choix. Ce qu'il est en mesure de faire avec l'écran de la figure 2.30.



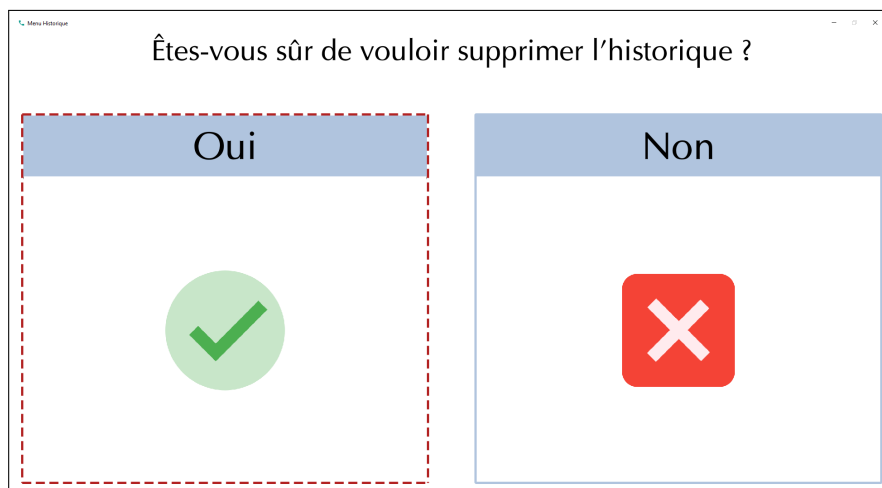


FIGURE 2.30 – Suppression des historiques d’appel

## 2.9 Validation utilisateur

Cette section décrit l’interview d’un utilisateur réalisée à la fin du stage.

### 2.9.1 Utilisateurs

La personne interviewée est atteinte de la maladie SLA (Sclérose latérale amyotrophique). Cette maladie provoque la paralysie progressive des muscles. Comme modalité d’interaction, elle utilisait son joystick de chaise roulante, afin de simuler une souris. Une ergothérapeute était également présente lors de cette interview.

### 2.9.2 Méthodologie

Cette interview s’est déroulée de la sorte : tout d’abord une période pendant laquelle l’utilisatrice a pris en main l’application, en réalisant certaines tâches définies.

Après cette partie d’observation, l’utilisateur a exprimé son ressenti à propos de l’application.

Enfin, l’ergothérapeute a également exprimé son avis à propos de l’application.

### 2.9.3 Observation

La première étape de l’interview a été d’observer l’utilisatrice effectuant quelques actions de bases avec l’application. L’utilisatrice n’avait jamais vu cette application auparavant.

La première action observée a été d’ouvrir la porte. Pour rappel, avec l’application qu’elle utilisait, il lui fallait au minimum sept clics pour ouvrir la porte (section 2.3). Lors de l’interview, elle a du utiliser deux actions (**Décrocher** sur la figure 2.17 et **Ouvrir la porte** sur la figure 2.18) afin de réaliser l’objectif. Ce qui fait un gain considérable par rapport à sept clics. Sa première réaction a été de louer la facilité avec laquelle elle est capable d’effectuer cette action. C’est la seule tâche où des

mesures ont été prises, car c'est la seule facilement comparable à la solution actuelle de l'utilisatrice.

La deuxième action observée a été l'affichage des détails d'un contact. Le but par là était de tester la navigation dans une liste, présentée à la section 2.7. L'utilisatrice a rapidement trouvé comment utiliser la liste, mais aurait préféré un affichage des contacts sous forme de cases. En effet, cela est dû au fait qu'elle utilise une souris comme modalité d'interaction.

L'utilisatrice s'est ensuite baladée dans l'interface, en découvrant par elle-même les fonctionnalités. Pour elle, tout était assez clair et surtout simple d'utilisation.

#### **2.9.4 Ressenti de l'utilisatrice et avis de l'ergothérapeute**

Pour l'utilisatrice, l'utilisation de l'application est intuitive et simple. Le seul point qui la dérangeait était le fait que la liste des contacts n'est pas représentée sous forme de cases.

Il est à noter que l'utilisatrice utilise toujours cette application à l'heure actuelle.

L'ergothérapeute, pour sa part, a trouvé l'interface vraiment intéressante, pas uniquement pour les personnes lourdement handicapées. Pour elle, cela pourrait également servir pour les personnes mal-voyantes, pour les personnes âgées, ou tout simplement pour les personnes à la recherche de quelque chose de simple. Elle a également insisté sur la demande qu'il existe pour une interface comme celle-là.

### **2.10 Conclusion**

Ce chapitre a présenté l'application conçue et développée durant le stage. Plusieurs points ont été abordés, comme les objectifs de l'application, la méthodologie, la conception de l'interface et des interactions, ou encore le résumé de l'interview d'une utilisatrice. À la fin de ce chapitre, une application de téléphonie pour les personnes handicapées est obtenue.

Le chapitre suivant portera un regard critique sur cette application, afin de pointer les différentes pistes d'améliorations possibles pour la suite. Ces pistes seront explorées au chapitre 5, à la suite de l'état de l'art (chapitre 4)



# Chapitre 3

## Regard critique sur le stage

Ce chapitre porte sur l'analyse critique de l'application créée durant le stage. L'analyse se porte d'une part sur la partie technique, c'est-à-dire sur le choix des bibliothèques et les choix posés pour la gestion des données. D'autre part, sur la partie interactions, qui comporte la gestion des interactions et l'interface graphique. Enfin, ce chapitre termine par la vérification de la complétude des objectifs initiaux de l'application VoIP.

### 3.1 Liblinphone et Qt

Les fonctions VoIP de Liblinphone fonctionnent sans aucun souci. La librairie est claire, simple d'utilisation. De plus, elle est open-source, ce qui signifie qu'il est tout-à-fait possible d'y apporter un changement. Un autre avantage est qu'elle est compatible Windows, Mac OS et Linux. Porter cette solution sur Mac OS et Linux est donc tout à fait possible.

Le constat est le même pour Qt, qui fonctionne également très bien. De même, Qt est compatible Windows, mais aussi Mac OS et Linux.

Les bibliothèques de VoIP et graphique sont donc une force de l'application développée.

### 3.2 Gestion des données

Il avait été décidé de créer des contacts propres à l'application. La principale raison de ce choix était la simplicité à mettre en place une solution fonctionnelle, dans l'optique de la réalisation d'un prototype. Cependant, plusieurs problèmes apparaissent : tout d'abord, l'utilisateur doit encoder la totalité de ses contacts à la main. Il aurait été possible de concevoir une fonctionnalité qui importe les contacts d'autres comptes, comme Google ou Microsoft. Ensuite, comme les données sont dans une base de données locale, si l'utilisateur possède deux ordinateurs différents, il doit encoder ses contacts sur les deux ordinateurs. De plus, d'un point de vue modélisation, la solution proposée n'est pas optimale. En effet, si un utilisateur possède une adresse SIP, un numéro de téléphone classique et un numéro de GSM, il est encodé trois fois dans la base de données. Cet aspect peut être corrigé facilement en

utilisant un standard de représentation des contacts, tel que vCard<sup>1</sup>.

La simplicité à mettre en place les contacts dans le cadre d'un prototype est une force. Cependant, des améliorations possibles sont la compatibilité avec un standard (vCard) et une gestion plus centralisée des contacts.

### 3.3 Interaction et interface

La technique de balayage mise en place ainsi que le moyen de naviguer dans la liste des contacts ont tous deux reçu de bonnes notes auprès de l'utilisateur interviewé, ainsi que de l'ergothérapeute présente lors de l'interview (section 2.9). Même constat pour ce qui est de l'interface.

Un autre bon point est la possibilité de l'interface d'être contrôlable par un grand panel de modalités d'interaction. En effet, il est possible d'utiliser l'application avec des boutons poussoirs, la souris, le clavier, les télécommandes, etc.

L'inconvénient est d'avoir tout dû ré-implémenter, et pas d'avoir utilisé un outil de balayage existant. Cela aurait permis d'utiliser un outil dont l'utilisateur avait déjà l'habitude, sans lui rajouter une nouvelle méthode à appréhender.

De plus, il a fallu ré-implémenter l'outil de lecture d'éléments. En effet, le lecteur d'éléments intégré à Windows ne fonctionnait pas correctement.

### 3.4 Objectifs

Cette section s'intéresse aux objectifs initiaux de l'application : quelles sont ceux remplis et ceux à remplir.

Le premier était d'avoir une interface graphique adaptée aux personnes à mobilité réduite. Elle devait donc être claire, épurée, intuitive et facile d'utilisation. L'interview de l'utilisateur (section 2.9) a bien validée cette interface.

Le deuxième objectif concernait les modalités d'interactions. L'interface devait être contrôlable par toutes les modalités d'interactions avec lesquels Soline est compatible, du moins la télécommande, le bouton et la voix, ainsi que le couple clavier/souris. L'application devait également être capable de se connecter au système Soline, pour l'avertir notamment en cas d'appel entrant. L'application est bien contrôlable par la télécommande, le bouton et le couple clavier/souris. De plus, toutes les modalités qui simulent un clic sur une touche du clavier (section 2.7) peuvent être utilisées pour contrôler l'application. En ce qui concerne la voix, l'objectif était d'utiliser le système Soline pour récupérer les commandes. Comme expliqué dans la section 2.7, le protocole conçu permet d'utiliser un dispositif de reconnaissance vocale externe, et donc Soline, afin de contrôler l'application par la voix. Cependant, le fonctionnement de cette partie n'a pas été testé.

L'application devait avoir toutes les fonctionnalités d'une application VoIP. Cet objectif est entièrement rempli. En effet, toutes les fonctionnalités VoIP fonctionnent correctement (appels, enregistrement de compte, etc).

---

1. <https://tools.ietf.org/html/rfc6350>

La quatrième objectif était d’avoir une gestion des contacts et des historiques, ce qui a été implémenté dans l’application développée.

Enfin, l’application doit être capable de s’exécuter au minimum dans un environnement Windows, ce qui est également le cas. Cependant, cet objectif n’est rempli qu’au minimum, dans le sens où l’application n’est disponible que pour Windows. A noter qu’un portage sur Mac OS ou Linux peut se faire assez rapidement.

## 3.5 Conclusion

Le tableau 3.1 résume les principales forces et améliorations possibles du point de vue technique de l’application conçue durant le stage. Le tableau 3.2 se focalise lui sur le point de vue utilisateur, c’est-à-dire sur l’interface et les interactions.

Forces	Améliorations possibles
Fonctions VoIP	Multi-plateforme Standard pour les contacts

TABLE 3.1 – Forces et améliorations possibles du point de vue technique

Forces	Améliorations possibles
Interface simple et intuitive Nombre de modalités gérées	Utilisation d’outils existants

TABLE 3.2 – Forces et améliorations possibles du point de vue utilisateur

Du point de vue techniques, plusieurs pistes d’améliorations sont possibles. La première serait d’ouvrir l’application aux autres systèmes d’exploitation (Mac OS et Linux), ainsi qu’aux mobiles (iOS, Android et Windows Phone). La seconde serait d’utiliser un standard pour la gestion des contacts, afin de pouvoir les importer facilement. Dans l’optique d’une solution globale qui contiendrait autre chose que simplement l’application de téléphonie, une amélioration serait de centraliser la gestion des contacts, afin qu’ils soient accessibles par plusieurs applications.

Du point de vue utilisateur, l’amélioration la plus importante serait d’utiliser les outils existants afin de permettre à l’utilisateur d’utiliser tous les outils auxquels il est habitué.

Le chapitre suivant présentera l’état de l’art en matière d’interactions pour les personnes lourdement handicapées, afin de trouver des pistes pour intégrer les outils existants à cette application et pour concevoir une taxonomie des interactions.



# Chapitre 4

## État de l'art

Ce chapitre traite des recherches effectuées en termes d'interactions pour les personnes ayant un handicap moteur conséquent, basées sur l'expérience acquise durant le développement de l'application dans le cadre du stage.

Pour communiquer avec un ordinateur, deux éléments sont nécessaires : un canal de communication, et une interface. Un canal de communication est une modalité d'interaction. Cela peut être le système moteur, la parole, la vue, le regard, etc. Il y a des modalités d'entrée (regard, parole, etc) ainsi que des modalités de sorties (la vue, l'audition, etc). La communication avec un ordinateur peut être multimodale, c'est-à-dire utiliser plusieurs canaux de communications. Le premier système multimodal est le *"Put-That-There"* de Richard Bolt [Bolt, 1980]. Ce système permet de positionner à l'écran différentes formes géométriques, et de les manipuler. Il fonctionne avec deux modalités d'entrée, la parole et les gestes. La parole est utilisée pour indiquer la commande à utiliser, et les gestes servent à indiquer à l'écran la localisation. L'interface est la façon de présenter l'information, d'organiser les éléments à l'écran.

La figure 4.1 représente le cycle d'interaction multimodale entre un humain et une machine tel que défini dans [Dumas *et al.*, 2009]. Lorsque l'humain veut communiquer avec un ordinateur, sa communication va passer par quatre états. Le premier est l'état de décision, dans lequel le contenu de la communication est préparé. Le second est l'état d'action, où la communication va être transmise via un canal de communication (modalité). Ensuite, la machine va percevoir la communication (par des capteurs par exemple), puis interpréter le contenu de la communication. Ensuite, la machine va construire, à partir du message reçu, une réponse à envoyer à l'utilisateur. Enfin, la quatrième étape de la machine va être de transmettre le message à l'utilisateur, de nouveau via un canal de communication (visuel, etc). Troisième état de la communication du point de vue de l'utilisateur, ce dernier va percevoir l'information transmise par la machine (lire à l'écran, entendre une notification, etc). Et enfin, l'utilisateur va interpréter l'information qu'il a reçue, éventuellement pour refaire le cycle, et transmettre un nouveau message. Dans le cadre de ce travail, nous allons aborder les moyens d'actions (façon de transmettre le message) du point de vue des utilisateurs (section 4.1) et du point de vue de la machine (sections 4.2 et 4.3).



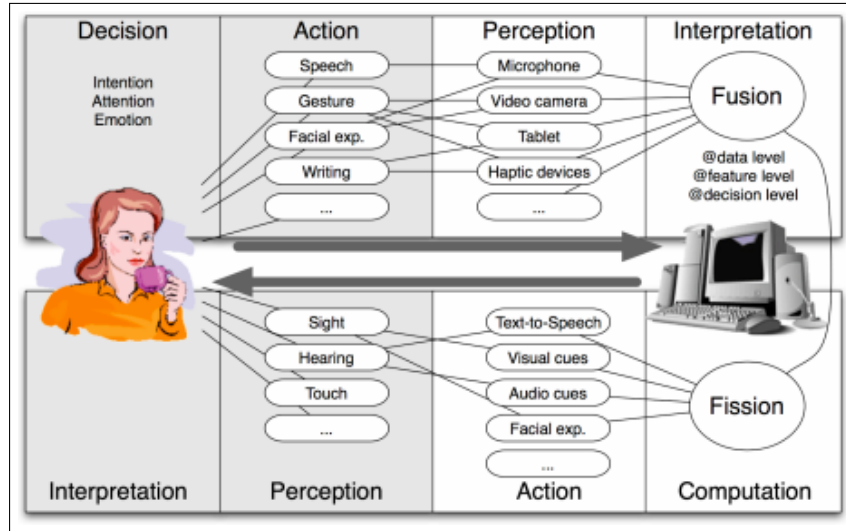


FIGURE 4.1 – Cycle d’interaction multimodale homme-machine [Dumas *et al.*, 2009]

## 4.1 Modalités d’interaction d’entrée

Cette section décrit toutes les modalités d’entrée disponibles pour les personnes ayant des capacités motrices réduites pour interagir avec un ordinateur. D’après [Polacek *et al.*, 2015], il existe quatre catégories majeures différentes : les modalités mécaniques, le regard, la parole et les signaux bio-électriques. Ces modalités correspondent aux canaux de communication.

### 4.1.1 Mécanique

La première modalité mécanique est le clavier de l’ordinateur. Les personnes à qui il reste des capacités motrices au niveau des mains peuvent l’utiliser. Pour une personne qui n’aurait plus qu’une main, se pose un premier problème : l’appui simultané. L’appui simultané sur plusieurs touches est utile notamment pour écrire une majuscule (MAJ + lettre) ou encore pour écrire le symbole € (ALT-GR + E). Ce problème peut être résolu en utilisant les touches rémanentes des systèmes d’exploitation. Les touches rémanentes permettent d’utiliser l’appui simultané en n’appuyant que sur une touche à la fois. Par exemple, pour écrire la lettre *E*, il faut appuyer sur la touche *maj*, et puis appuyer sur la touche *e*. Il n’est plus nécessaire d’appuyer sur les deux touches en même temps.

Il existe également toutes sortes de claviers optimisés, comme par exemple les claviers ergonomiques Maltron (Exemple : figure 4.2). Ces claviers sont adaptés aux personnes avec des capacités motrices réduites, qui peuvent utiliser au moins une main.



FIGURE 4.2 – Clavier Maltron à une main (droite)

Pour les personnes qui ont un handicap plus sévère, et qui ne sont pas capables d'utiliser les premières solutions présentées, il existe toutes sortes de dispositifs qui, associés aux méthodes de balayages (qui seront présentées dans la section suivante), permettent de contrôler un ordinateur. Ces dispositifs fonctionnent comme des interrupteurs. Dans leurs recherches, Stavroula Ntoa et ses collègues [Ntoa *et al.*, 2013] présentent une liste non exhaustive de ce qu'il existe comme dispositifs d'interrupteurs. En fonction de l'handicap d'une personne, il est donc possible de choisir le meilleur dispositif pour la situation de la personne. Voici les différents dispositifs [Ntoa *et al.*, 2013] :

**Le bouton** est activé grâce à une pression sur celui-ci. Il est relié à l'ordinateur via une interface spéciale.



FIGURE 4.3 – Exemple de bouton <sup>1</sup>

**Le joystick** peut être activé sous neuf états différents : les huit directions et en cliquant dessus.



FIGURE 4.4 – Exemple de joystick <sup>2</sup>

**L'interrupteur à poigne** est un interrupteur mécanique qui est porté dans la paume de la main de l'utilisateur. Pour l'activer, il lui suffit de l'écraser ou de le pincer.



FIGURE 4.5 – Exemple d'interrupteur à poigne <sup>3</sup>

---

1. <https://www.ablenetinc.com/technology/switches/big-beamer-slat-receiver>

2. <https://images.dlf.org.uk/mee/products/full/0105619.jpg>

3. <https://www.enablemart.com/media/catalog/product/cache/3/image/700x700/17f82f742ffe127f42dca9de82fb5>

**L'interrupteur à levier** est un dispositif mécanique léger, composé d'une petite barre qui peut se positionner soit d'un côté, soit de l'autre. Le levier s'active lorsque l'utilisateur change le levier de position.



FIGURE 4.6 – Exemple d'interrupteur à levier<sup>4</sup>

**L'interrupteur à ruban** est un interrupteur qui peut être activé en le pliant dans n'importe quelle direction.

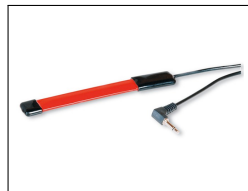


FIGURE 4.7 – Exemple d'interrupteur à ruban<sup>5</sup>

**L'interrupteur à feuille** est, dans sa forme, assez similaire à l'interrupteur à levier. En effet, il possède également une barre. Seulement, celle-ci ne reste que dans la position de base. Pour l'activer, il faut déplacer d'un côté ou de l'autre la barre, qui se remet par la suite dans sa position initiale.

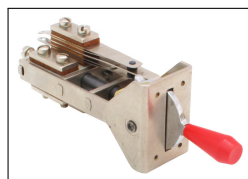


FIGURE 4.8 – Exemple d'interrupteur à feuille<sup>6</sup>

---

4. <http://www.directindustry.com/prod/apem/product-7593-519452.html>

5. <http://www.inclusive.co.uk/Img/Dyn/Cache/Products/1976-271112094958-1514609478.jpg>

6. <https://www.surplussales.com/Images/Switches/Leaf/swl-harris-lg.jpg>

**L'interrupteur au pouce** est un dispositif qui se tient dans la main, et qui est activable avec un appui du pouce sur un bouton.



FIGURE 4.9 – Exemple d'interrupteur au pouce<sup>7</sup>

**L'interrupteur au doigt** est un dispositif porté au doigt, et qui s'active grâce à une petite pression.



FIGURE 4.10 – Exemple d'interrupteur au doigt<sup>8</sup>

**L'interrupteur à ficelle** est un interrupteur qui s'active en tirant sur une ficelle.



FIGURE 4.11 – Exemple d'interrupteur à ficelle<sup>9</sup>

---

7. <https://enablingdevices.com/images/products/large/04abac37-e30d-4dd0-9eef-e64500502d72.jpg>

8. <https://enablingdevices.com/images/products/large/de5465e2-2b4e-4d67-a1ab-730b27c641bf.jpg>

9. <http://www.zytech.com.au/uploads/Files/Access/switches/string-switch.jpg>

**L'interrupteur à pédale** s'active avec les pieds. Cela permet de retirer toute interaction avec les mains.



FIGURE 4.12 – Exemple d'interrupteur à pédale<sup>10</sup>

**L'interrupteur à pression d'air** s'active en compressant le dispositif, notamment avec les mains, le pied ou encore la tête.



FIGURE 4.13 – Exemple d'interrupteur à pression d'air<sup>11</sup>

**L'interrupteur pneumatique** s'active en soufflant dans le dispositif. Cela permet de contrôler son ordinateur avec la respiration.



FIGURE 4.14 – Exemple d'interrupteur pneumatique<sup>12</sup>

---

10. [http://www.scythe-eu.com/uploads/tx\\_cfamooflow/USB-2Foot-Switch2-View2\\_03.jpg](http://www.scythe-eu.com/uploads/tx_cfamooflow/USB-2Foot-Switch2-View2_03.jpg)

11. <http://www.oneswitch.org.uk/IMAGES/1/AGS/switch-ultra-light.jpg>

12. <http://www.hirstwood.com/wp-content/uploads/2014/08/Pneumatic.jpg>

**L'interrupteur infrarouge** s'active sans contact, notamment en passant la main devant. Il fonctionne en détectant un faisceau de lumière infrarouge réfléchi.



FIGURE 4.15 – Exemple d'interrupteur infrarouge<sup>13</sup>

**L'interrupteur au menton** est un interrupteur mécaniques monté sur un collier, à mettre autour du cou. Il peut y avoir plusieurs interrupteurs sur un seul collier.



FIGURE 4.16 – Exemple d'interrupteur au menton<sup>14</sup>

**L'interrupteur à morsure** est un interrupteur qui s'active lorsque l'utilisateur mord celui-ci.



FIGURE 4.17 – Exemple d'interrupteur à morsure<sup>15</sup>

---

13. <https://www.ablenetinc.com/technology/switches/scatir-switch-deluxe-kit>

14. <https://www.ablenetinc.com/chin-switch>

15. <https://enablingdevices.com/images/products/standard/ebc035dd-9fdc-4920-ac3f-e7b76c0ec1a3.jpg>

**L'interrupteur à capteur de proximité** est activé lorsque l'utilisateur approche une partie de son corps à 10mm de l'interrupteur. Il peut s'agir de sa main, de son pied, de son coude, etc.



FIGURE 4.18 – Exemple d'interrupteur à capteur de proximité<sup>16</sup>

### Dispositifs de pointage

En plus des dispositifs d'interrupteur, il existe des dispositifs de pointage pour les personnes atteintes de handicap moteur. Ces dispositifs servent à simuler une souris. Premièrement, il y a le trackball (figure 4.19), un dispositif avec une roulette qui permet de bouger le curseur de la souris en ne bougeant que la roulette.



FIGURE 4.19 – Exemple de trackball<sup>17</sup>

Deuxièmement, il existe les dispositifs de *head tracking* (figure 4.20), c'est-à-dire des dispositifs qui détectent les mouvements de la tête.



FIGURE 4.20 – Exemple de *head tracking*<sup>18</sup>

Cependant, un problème se pose avec les dispositifs de pointage : l'absence de mécanisme de sélection. En effet, l'utilisateur est capable de déplacer le curseur à l'écran, mais incapable de sélectionner un élément. Pour répondre à ce problème,

---

16. <https://www.ablenetinc.com/technology/switches/candy-corn-proximity-sensor-switch>

17. <https://www.ablenetinc.com/bigtrack-trackball-switch-adapted>

18. <http://www.naturalpoint.com/smarnav/images/products/smarnav4-sideangle.jpg>



deux méthodes sont envisageables [Polacek *et al.*, 2015] : le *dwell-time*, et l'interaction multimodale.

La méthode du *dwell-time* consiste à sélectionner un élément lorsque l'utilisateur ne bouge plus son curseur pendant un certain temps. Cela lui permet ainsi de sélectionner un élément à l'écran. Néanmoins, un autre problème se pose : lorsque l'utilisateur regarde quelque chose à l'écran, il ne veut pas forcément qu'une action se déclenche. Ce problème est celui du *Midas Touch* [Jacob, 1991] (ou toucher de Midas, en référence au héros du même nom dans la mythologie grecque, qui transformait en or tout ce qu'il touchait). Deux solutions permettent de résoudre cette difficulté. La première est proposée par Margrit Betke et ses collègues. Cette solution consiste à ajouter à l'écran une zone de repos pour le curseur [Betke *et al.*, 2002]. Placé dans cette zone, le curseur ne pourra plus activer une sélection. La deuxième est proposée par Zich L. et consiste à afficher une fenêtre *pop-up* à la fin du temps défini pour la méthode du *dwell-time*.

L'interaction multimodale consiste à utiliser une modalité d'interaction différente pour la sélection. Quelques recherches ont été menées pour écrire au clavier avec cette méthode. Notamment, un système qui fonctionne avec le regard comme dispositif de pointage, et les dents comme fonction de sélection [Zhao *et al.*, 2012]. La reconnaissance vocale ainsi que le langage non-verbal sont deux autres exemples de modalités qui sont utilisées avec les dispositifs de pointage.

### 4.1.2 Parole

Il est possible d'interagir avec l'ordinateur en utilisant la parole, cela grâce à des systèmes de reconnaissance vocale. Il suffit de prononcer un ordre compréhensible par l'ordinateur à celui-ci, afin que ce dernier effectue une action correspondante.

Andrew Sears et ses collègues ont menés une étude afin de mesurer la différence de performance et de satisfaction entre les personnes atteintes d'un handicap moteur et les personnes non atteintes [Sears *et al.*, 2001]. Pour cette étude, les auteurs de l'article ont demandé à deux groupes de personnes (d'un côté les personnes atteintes d'un handicap moteur, et de l'autre côté le reste des personnes) de dicter un texte.

Concernant la performance, les auteurs ont mesuré les erreurs de texte, le nombre de mots écrits par minutes et la qualité du document final. Il s'avère que les deux groupes obtiennent les mêmes performances lors de la diction d'un texte. Ce qui est un bon point pour la modalité de la parole pour les personnes lourdement handicapées.

Par contre, du point de vue de la satisfaction, les auteurs ont remarqué une forte différence entre les deux groupes. Le groupe des personnes handicapées s'est montré beaucoup plus positif à utiliser un tel système. En effet, pour l'autre groupe, l'utilisation du clavier et de la souris reste plus facile et rapide que l'utilisation de la reconnaissance vocale.

### 4.1.3 Regard

Lorsque la personne n'est plus en mesure d'utiliser ni ses mains ni sa parole mais qu'elle peut toujours se servir de ses yeux (c'est le cas par exemple des personnes

atteintes du locked-in syndrome), il est possible d'utiliser le regard comme modalité d'interaction.

Le locked-in syndrome est une maladie neurologique qui se caractérise par une quadriplégie (paralysie des quatre membres), une anarthrie (trouble de la parole) ainsi que la conservation du mouvement des yeux.

Les interactions avec le regard fonctionnent grâce à un dispositif de suivi des yeux. Ce dispositif calcule le point visé en mesurant la position des yeux. Le regard est utilisé comme un dispositif de pointage (simulation de la souris). Comme expliqué dans la section 4.1.1, les dispositifs de pointage font face au problème d'absence de sélection, qui sont résolus par le *dwell-time* et l'interaction multimodale.

#### 4.1.4 Signaux bio-électriques

Les signaux bio-électriques sont une quatrième modalité d'interaction pour personnes lourdement handicapées. Il existe deux techniques non-invasives pour l'utilisateur : la contraction des muscles intentionnelle et l'interaction cerveau-machine.

La contraction des muscles intentionnelle est mesurée grâce à l'électromyographie, qui permet de détecter l'activité des muscles. En 2002, Felzer et ses collègues [Felzer et Freisleben, 2002] ont imaginé un système de contrôle de chaise roulante sans les mains. Ce système permettait à l'utilisateur de conduire sa chaise roulante à l'aide de n'importe quels muscles choisis par l'utilisateur. Les électrodes servant à mesurer l'activité des muscles se placent à même la peau, à l'endroit des muscles choisis. Cette méthode est assez peu contraignante pour l'utilisateur.

L'interaction cerveau-machine fonctionne en mesurant les signaux électriques directement sur la tête. Cela se fait à l'aide de l'électroencéphalographie, qui permet de détecter l'activité électrique du cerveau. Cette technique est plus onéreuse que celle de la contraction des muscles intentionnelle, et les signaux captés sont plus difficiles à traiter.

## 4.2 Balayage d'éléments

Cette section décrit la méthode de balayage d'éléments, qui permet à un utilisateur de contrôler une interface avec un minimum de modalités. Nous allons premièrement aborder les différentes techniques qu'il existe, et ensuite les différents moyens de mettre en œuvre cette technique.

### 4.2.1 Techniques

Le balayage d'éléments est une méthode d'interaction à destination des personnes ayant un handicap moteur conséquent. Ces dernières, grâce à cette méthode, peuvent contrôler une interface avec au minimum un signal binaire. C'est-à-dire un bouton, un point sur l'écran à fixer du regard, un son à produire, etc.

Le balayage consiste à parcourir les éléments de l'écran séquentiellement, et ensuite sélectionner un élément pour l'activer. En général, l'élément actif est mis en surbrillance.

Dans leurs recherches, Stavroula Ntoa et ses collègues distinguent cinq catégories de balayage [Ntoa *et al.*, 2013] :

- Le balayage par bloc [Biswas et Robinson, 2008], [Ntoa *et al.*, 2004]
- Le balayage bi-directionnel [Ntoa *et al.*, 2013]
- Le balayage octo-directionnel [Biswas et Robinson, 2008], [Ntoa *et al.*, 2013]
- Le balayage par hiérarchie [Ntoa *et al.*, 2013]
- Le balayage par *cluster* [Biswas et Robinson, 2008]

Outre ces catégories, un balayage peut être adaptatif, c'est-à-dire qu'il peut adapter la vitesse de parcours des éléments en fonction des performances de l'utilisateur.

### Balayage par bloc

Dans la technique du balayage par bloc, l'écran est divisé en plusieurs zones, elles-mêmes divisées en sous zones plus petites. La figure 4.21 représente cette technique. Pour sélectionner l'élément en gris sur la figure 4.21, l'utilisateur doit d'abord sélectionner la grande zone en haut à gauche, puis la zone moyenne en bas à droite, et enfin la petite zone en bas à droite.

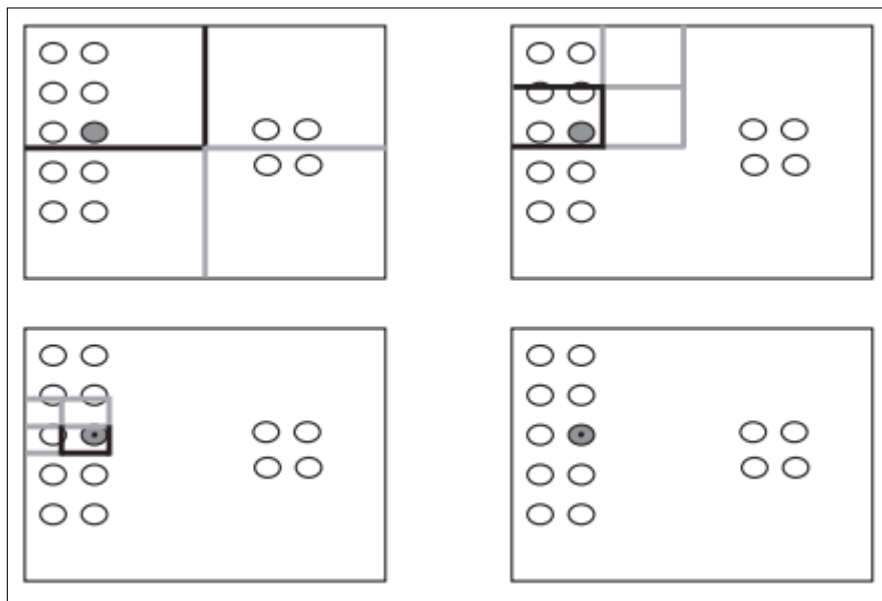


FIGURE 4.21 – Balayage par bloc [Biswas et Robinson, 2008]

Le parcours séquentiel s'effectue en passant de zone en zone. Il existe plusieurs façons d'arranger les zones [Ntoa *et al.*, 2013], notamment en utilisant un arrangement en lignes/colonnes. Dans celui-ci, les éléments sont groupés en lignes, puis en colonnes. La haut de la figure 4.22 représente cet arrangement, tandis que le bas représente un arrangement colonnes/lignes, qui groupe les éléments en colonnes, puis en ligne. La technique de lignes/colonnes est fortement répandue pour les claviers virtuels [Ntoa *et al.*, 2013].

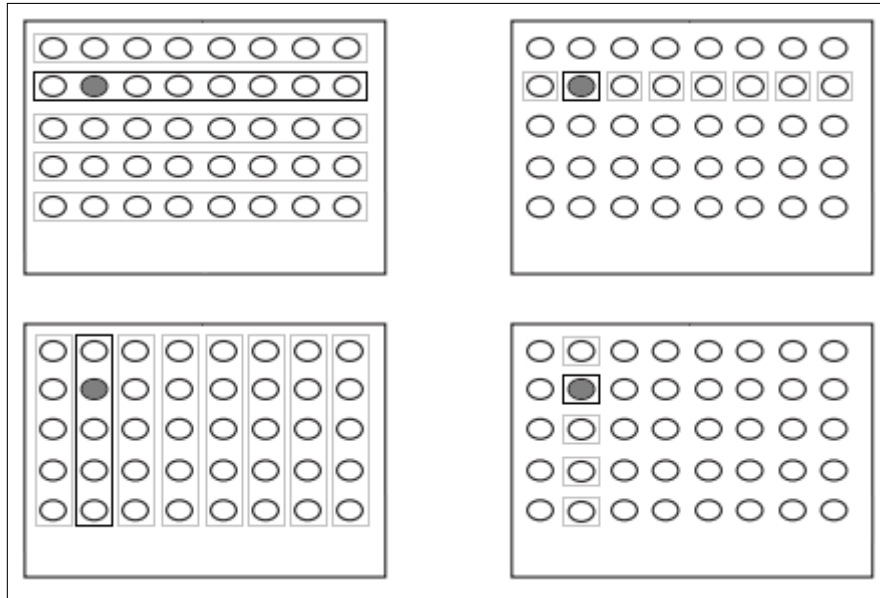


FIGURE 4.22 – Balayage par bloc en lignes/colonnes et colonnes/lignes

Un exemple est le clavier visuel inclus dans Windows, comme montré par la figure 4.23

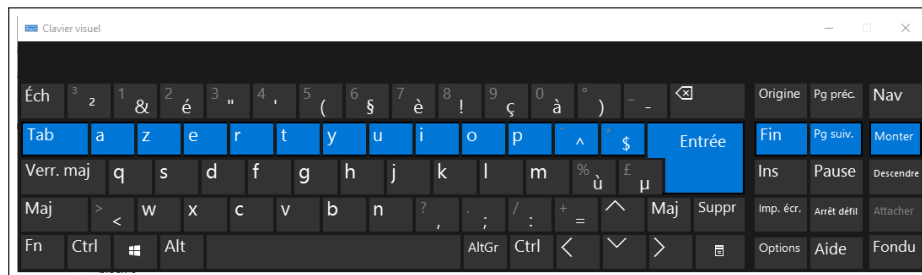


FIGURE 4.23 – Balayage par bloc en lignes/colonnes : clavier visuel Windows

## Balayage bi-directionnel

Le balayage bi-directionnel est une technique qui permet de sélectionner un élément en se servant des coordonnées X et Y de l'écran. En premier lieu, une ligne horizontale parcourt l'écran de haut en bas. Lorsque la barre survole l'élément voulu, l'utilisateur appuie sur une touche et un pointeur passe de gauche à droite le long de la ligne horizontale. L'utilisateur doit appuyer de nouveau lorsque le pointeur arrive sur l'élément voulu. La figure 4.24, tirée du logiciel Cross-Scanner [Cooper et Associés, 2012], illustre cette technique. Cross-Scanner est un logiciel qui implémente le balayage bi-directionnel.

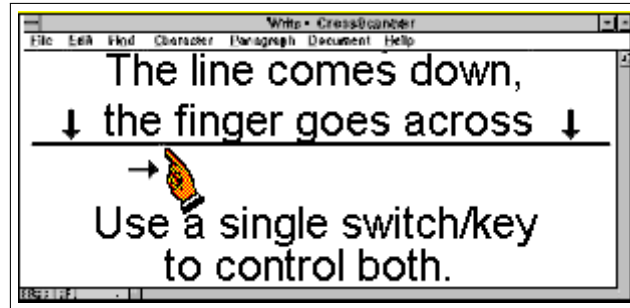


FIGURE 4.24 – Balayage bi-directionnel : Cross Scanner [Cooper et Associés, 2012]

### Balayage octo-directionnel

Dans la technique du balayage octo-directionnel, l'utilisateur a la possibilité de déplacer le curseur de la souris dans huit directions : en haut, en bas, à gauche, à droite, en haut à gauche, en haut à droite, en bas à gauche et en bas à droite. Le curseur de sélection se déplace sur les huit choix de directions. Lorsque l'utilisateur sélectionne la direction, le curseur de la souris se déplace sur une ligne dans la direction demandée. L'utilisateur doit indiquer au système lorsque le curseur doit s'arrêter. La figure 4.25 représente le balayage octo-directionnel. Le curseur doit descendre en bas à gauche, puis aller à gauche pour sélectionner l'élément voulu.

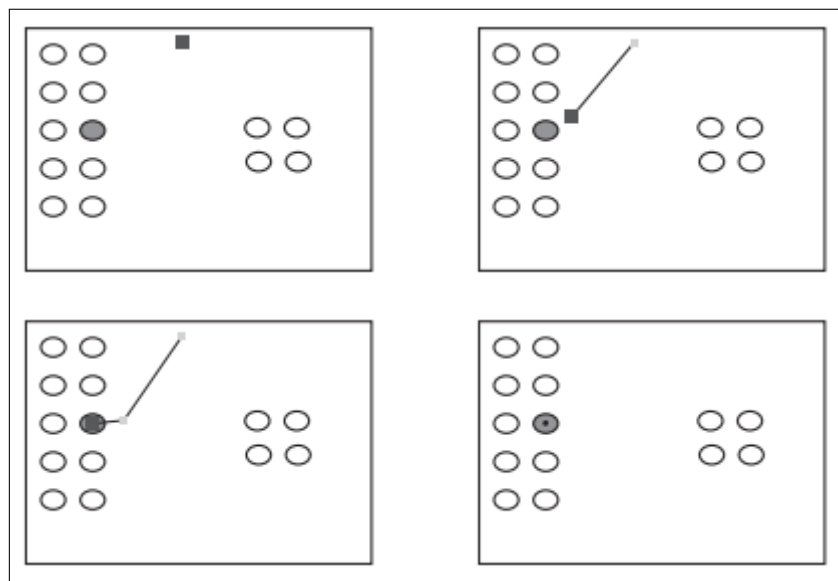


FIGURE 4.25 – Balayage octo-directionnel [Biswas et Robinson, 2008]

### Balayage par hiérarchie

La technique de balayage par hiérarchie est une technique où l'accès aux éléments est déterminé en fonction de leur position dans la hiérarchie de la fenêtre. Pour accéder à un bouton du menu, il faut d'abord accéder au menu.

### Balayage par *cluster*

Dans la technique de balayage par *cluster*, les éléments de l'écran sont rassemblés en groupe en fonction de leur localisation à l'écran. La technique est illustrée par la

figure 4.26.

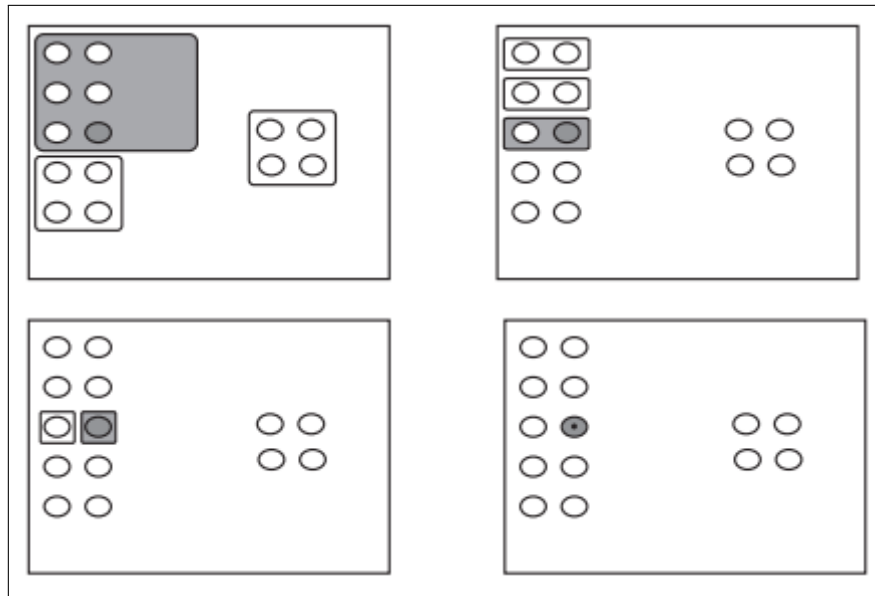


FIGURE 4.26 – Balayage par *cluster* [Biswas et Robinson, 2008]

### Balayage adaptatif

Un balayage dit adaptatif est un balayage qui est capable d'adapter la vitesse de parcours des éléments, en fonction des performances de l'utilisateur. Un balayage adaptatif par bloc a été proposé par Richard C. Simpson et Heidi Horstmann Koester [Simpson et Koester, 1999] et ensuite testé [Simpson *et al.*, 2006]. Le résultat de ces études est qu'un balayage adaptatif n'augmente pas la facilité d'utilisation.

## 4.2.2 Applications

Cette sous-section explique les différentes possibilités pour mettre en œuvre les techniques de balayage d'écran. Elle s'appuie sur les recherches de Stavroula Ntoa et ses collègues [Ntoa *et al.*, 2013]. D'après eux, il existe trois types de moyens pour cela : les applications qui ont leur propre système de balayage intégré, les bibliothèques de développement qui permettent de développer une application avec un système de balayage, et un outil plus générique qui propose une solution indépendante de l'application utilisée.

### Systèmes de balayage embarqués : navigateurs web

Les applications les plus pertinentes ayant un système de balayage embarqué sont les systèmes d'entrée de texte dans un éditeur spécial et les navigateurs internet. Les systèmes d'entrée de texte seront présentés dans la section 4.3.

Le web est aujourd'hui extrêmement utilisé dans la vie courante, que ce soit pour les informations, pour le loisir, pour les commerces en ligne ou encore pour communiquer. Il est donc important de fournir un accès équitable à tout le monde. C'est ce but que suivent les applications présentées ci-dessous.

#### **Le navigateur web AVANTI** [Stephanidis *et al.*, 1998]

Le navigateur web AVANTI fait partie du projet AVANTI, qui a pour but de permettre à l'utilisateur d'interagir avec une multitude de dispositifs d'entrée et sortie différents. Le projet AVANTI est spécialement conçu pour les personnes ayant un handicap moteur, et pour les personnes aveugles. De plus, le système est capable de s'adapter dynamiquement aux capacités de l'utilisateur. A noter que le navigateur suit la méthodologie de conception d'interface unifiée [Stephanidis et Savidis, 2002] (comme tous les autres éléments du projet AVANTI), qui permet de n'avoir qu'une seule interface à appréhender.

#### **MultiWeb** [Owens et Keller, 2000]

MultiWeb est un navigateur qui offre la possibilité à l'utilisateur de choisir son interface en fonction de son dispositif. Ce système propose six interfaces différentes :

1. Interface par défaut (clavier OU souris) : L'interface est contrôlable soit par le clavier (avec des raccourcis) soit par la souris (avec de gros boutons).
2. Interface contrôlable par interrupteur(s) : L'interface est contrôlable avec des interrupteurs grâce aux techniques de balayage d'écran.
3. Interface pour écran tactile : Les boutons sont plus grands, pour plus facilement cliquer dessus.
4. Interface pour clavier ET souris : Le clavier est utilisé pour naviguer dans le document et la souris pour cliquer sur les boutons.
5. Interface pour clavier uniquement : Tous les boutons sont remplacés par des raccourcis
6. Interface par menus : Le document est séparé en plusieurs grâce à des menus.

Il faut savoir que les auteurs de ce système ont impliqué directement des utilisateurs handicapés dans la conception de leur système.

**ARGO** [Ntoa et Stephanidis, 2005]

ARGO est un navigateur internet spécialement conçu pour les personnes mal-voyantes et les personnes ayant un handicap moteur des membres supérieurs. Pour ces derniers, le système prévoit un balayage d'écran, qui fonctionne à l'aide de trois interrupteurs. Ce balayage est un balayage par bloc (sous-section 4.2.1).

**KeySurf** [Spalteholz *et al.*, 2008]

Le navigateur KeySurf est contrôlable uniquement avec un clavier. La navigation se fait par raccourci dans le document. Néanmoins, ce navigateur peut être contrôlé par des interrupteurs, en proposant un balayage sur un clavier virtuel. Cela permet à l'utilisateur de sélectionner les raccourcis sur ce clavier, en utilisant les interrupteurs.

**FireScanner** [Ntoa *et al.*, 2009]

FireScanner n'est pas un navigateur, mais une extension pour le navigateur Firefox. Cette extension s'occupe de scanner la page que l'utilisateur consulte, pour repérer les balises HTML, et mettre en place un balayage par bloc (sous-section 4.2.1). Il est à noter que ce processus n'impacte pas le temps de chargement de la page web.

L'avantage de ces applications est de fournir directement une solution permettant le balayage par interrupteur aux personnes ayant un handicap moteur. Cependant, l'utilisateur devra s'équiper de plusieurs applications pour réaliser toutes ces tâches. Ce qui implique qu'il devra utiliser plusieurs méthodes de balayage différentes. Cela lui ajoute donc une difficulté supplémentaire étant donné qu'il devra s'habituer à toutes ces méthodes.

## Librairies

Afin de proposer des méthodes plus universelles de gestion du balayage, quelques recherches ont été lancées dans la conception de librairies, à destination des développeurs, pour concevoir des interfaces de balayage.

Les premiers à avoir essayé de concevoir une librairie sont Anthony Savidis et ses collègues [Savidis *et al.*, 1997]. Ces derniers ont développé un ensemble d'objets permettant d'augmenter les fonctionnalités des fenêtres Windows en permettant le support du balayage. Cependant, cette librairie est obsolète depuis les dernières versions de Windows.

La deuxième librairie a été créée, en 2003, par Constantine Steriadis et ses collègues [Steriadis et Constantinou, 2003]. Elle s'adresse particulièrement à la création d'interfaces pour les personnes quadriplégiques. Pour concevoir une interface à l'aide de cette librairie, il suffit d'utiliser le composant qu'ils ont créé : WIFSID (*Widget For Single-switch Input Devices*). Comme son nom l'indique, ce composant est destiné à être utilisé avec un seul interrupteur. Il est capable de remplir trois fonctions : mettre en évidence un élément lors du balayage, retirer cette mise en évidence de l'objet et enfin recevoir une entrée de l'interrupteur.

L'absence d'un réel intérêt pour ces librairies explique ce faible nombre de recherches liées. En effet, pour être utilisée, une librairie doit être maintenue, adaptée aux nouveaux systèmes d'exploitations, etc. Tout cela coûte énormément de ressources, ce qui explique ce faible intérêt.



## Outil indépendant

Afin de contrer l'inconvénient des systèmes de balayage embarqués, il existe les solutions indépendantes des applications. Un avantage de ces solutions est qu'elles fonctionnent sur tout le système, l'utilisateur pourra donc se spécialiser dans une seule solution de balayage, et pas une pour chaque application qu'il utilise. De plus, ces solutions permettent d'avoir un balayage sur une application qui n'est pas prévue pour cela, et donc de pouvoir utiliser n'importe quelle application. Voici ci dessous une liste des recherches effectuées sur le sujet :

### **CrossScanner** [Cooper et Associés, 2012]

CrossScanner est un outil qui permet d'avoir accès au balayage sur toutes les fenêtres Windows et Mac OS. Le balayage de CrossScanner est de type bi-directionnel (sous-section 4.2.1).

### **FastScanner** [Ntoa *et al.*, 2004]

FastScanner fournit un balayage sur toutes applications Windows. Cela se fait en récupérant les différents composants affichés sur les fenêtre.

### **ScanBuddy** [Factors, 2017]

ScanBuddy est un logiciel d'émulation de la souris. La première étape est de sélectionner la zone où l'action de la souris doit être effectué. La deuxième est de choisir l'action de la souris, grâce à un balayage bi-directionnel (sous-section 4.2.1) sur une interface de gestion de la souris. Cette interface contient les actions que la souris peut effectuer (double clic, clic droit, etc).

### **SwitchXS** [Instruments, 2017]

SwitchXS est un logiciel d'émulation du clavier et de la souris, pour Mac OS. Son interface se présente comme un clavier virtuel, avec en plus les actions de la souris disponibles. Un balayage par bloc (sous-section 4.2.1) s'effectue sur l'interface, afin de sélectionner l'action désirée.

### **Autonomia** [Steriadis et Constantinou, 2002]

Autonomia est un système de balayage d'écran qui permet de simuler une souris, d'écrire du texte et de lancer d'autres applications. Son fonctionnement est similaire aux autres logiciels d'émulation de clavier ou souris.

## 4.3 Entrée de texte

Cette section décrit les méthodes d'entrée de texte, ou claviers virtuels, pour les personnes ayant un handicap moteur. Cette section est inspirée de l'article de Ondrej Polacek et ses collègues [Polacek *et al.*, 2015]. Nous aborderons en premier lieu les moyens de sélection d'une lettre, et ensuite l'arrangement des caractères sur le clavier. Le but premier de l'étude des méthodes d'entrée de texte est de maximiser le débit d'écriture, c'est-à-dire le nombre de mots écrits par minute.

### 4.3.1 Sélection des caractères

La sélection des caractères est la manière de choisir la lettre à écrire. Sur un clavier d'ordinateur, la plupart des caractères sont sélectionnés en appuyant sur la touche correspondante. A l'exception notamment des lettres majuscules, où l'appui sur deux touches simultanément est nécessaire.

Pour les personnes invalides, il est très difficile d'utiliser un clavier. Une solution est de réduire le nombre de touches pour rendre le dispositif ou le clavier virtuel plus petit.

Les auteurs de l'article [Polacek *et al.*, 2015] ont extrait trois catégories de sélection : directe, par pointeur et par balayage.

#### Directe

La sélection directe permet aux utilisateurs de sélectionner un caractère en appuyant sur une ou plusieurs touches. L'utilisation du clavier normal d'un ordinateur se fait en sélection directe. Trois types de claviers sont disponibles pour la sélection directe : les claviers simultanés, les claviers ambigus et les claviers à encodage.

**Clavier simultané** Le clavier simultané permet de sélectionner un caractère en appuyant sur plusieurs touches. Cela permet de réduire le nombre de touches nécessaire, mais demande une capacité à appuyer sur plusieurs touches en même temps. De ce fait, ces claviers sont très peu utilisés dans le contexte des personnes invalides. La figure 4.27 illustre un tel clavier. En appuyant sur la touche 1 (K1), on obtient le caractère 'A'. En appuyant sur la touche 2 (K2), c'est le caractère 'B' qui est sélectionné. En appuyant sur les deux touches en même temps, on sélectionne le caractère 'C'.

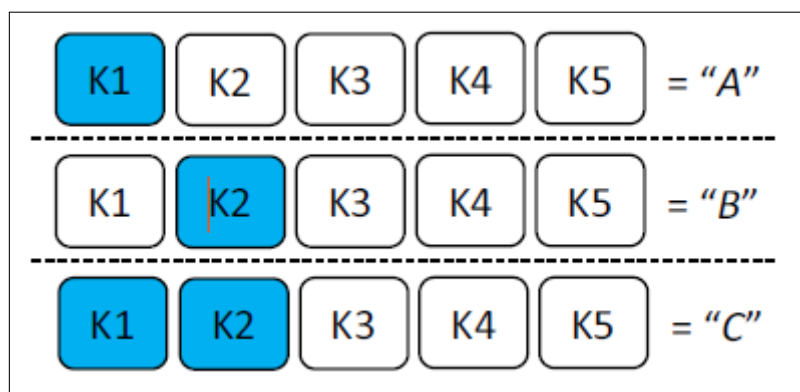


FIGURE 4.27 – Clavier simultané à 5 touches [Polacek *et al.*, 2015]

**Clavier ambigu** Un clavier ambigu est un clavier où plusieurs caractères sont associés à une seule touche. Il est illustré par la figure 4.28. Le clavier ambigu est fort utilisé par de nombreux téléphones portables .

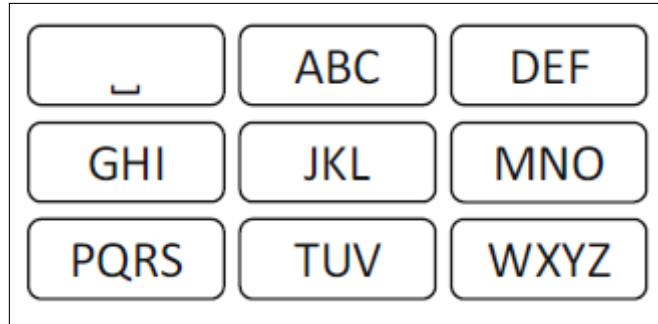


FIGURE 4.28 – Clavier ambigu [Polacek *et al.*, 2015]

**Clavier à encodage** Le clavier à encodage est le dernier type de clavier à sélection directe. Chaque caractère peut être choisi à travers une combinaison de touches unique. Le code Morse est un exemple d’encodage des caractères, et peut être utilisé pour un clavier à encodage avec deux touches. Le clavier à encodage est illustré par la figure 4.29.

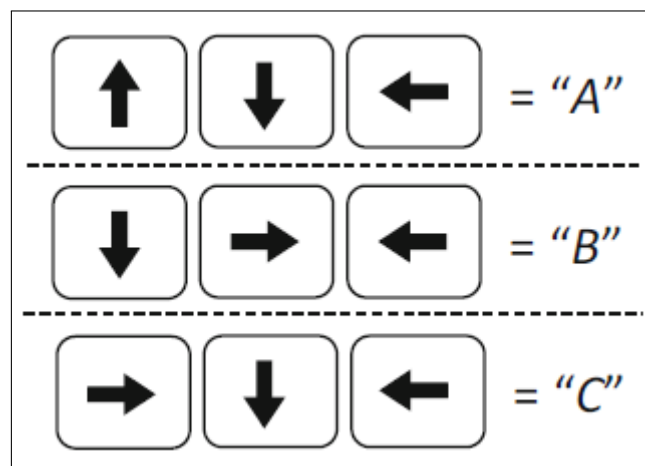


FIGURE 4.29 – Clavier à encodage avec 4 touches [Polacek *et al.*, 2015]

## Pointeur

La sélection par pointeur utilise un dispositif de pointage (sous-section 4.1.1) afin de sélectionner un caractère sur un clavier. Comme expliqué dans la section à propos des dispositifs de pointage, le souci est l’absence de mécanisme de sélection. Ce souci est contourné avec le méthode du *dwel-time* et de l’interaction multimodale.

## Balayage

La sélection par balayage permet de sélectionner un caractères à l’aide d’une technique de balayage. Ces techniques de balayage sont présentées dans la section 4.2.1.

### 4.3.2 Arrangement des caractères

Afin de maximiser le nombre de mots écrits par minute, certains chercheurs se sont penchés sur l'arrangement des lettres sur le clavier. Pour trouver un clavier optimal, il faut prendre en compte la fréquence des mots écrits. Seulement, la fréquence des mots dépend évidemment de la langue, et donc un clavier se doit d'être adapté à la langue de l'utilisateur. Une idée pour avoir un clavier plus universel a été de créer un arrangement sur base de l'alphabet, indépendamment donc de la langue. Cependant, chaque langue contient ses propres symboles, ce qui complexifie la recherche d'un clavier universel.

Il existe deux catégories d'arrangement des caractères [Boissiere et Dours, 2003] : statique et dynamique. Ces catégories font référence à la distribution des caractères, c'est-à-dire la représentation visuelle des caractères et les opérations à effectuer pour sélectionner le caractère. La différence entre les deux catégories est que les opérations à effectuer restent les mêmes au cours de l'écriture pour la catégorie statique, et changent pour la catégorie dynamique.

L'avantage de la distribution statique est qu'elle demande moins d'efforts intellectuels que la distribution dynamique. En effet, l'utilisateur peut mémoriser la position des caractères une seule fois pour la distribution statique.

La distribution dynamique adapte automatiquement les opérations à effectuer en fonction du contexte actuel [Boissiere et Dours, 2003]. Cela permet de minimiser le nombre de clics nécessaire pour écrire un mot.

Les figures 4.30 et 4.31 illustrent la différence entre les deux distributions. L'exemple est la façon d'écrire le mot "Je". Avec la distribution statique, il faut appuyer une fois sur le "JKL", et deux fois sur le "DEF". Tandis que pour la distribution dynamique, la touche "DEF" se transforme en "EDF" étant donné qu'il est plus probable de trouver un "E" que un "D" ou un "F" après un "J". Il faut donc n'appuyer qu'une seule fois sur le "EDF".



FIGURE 4.30 – Distribution statique

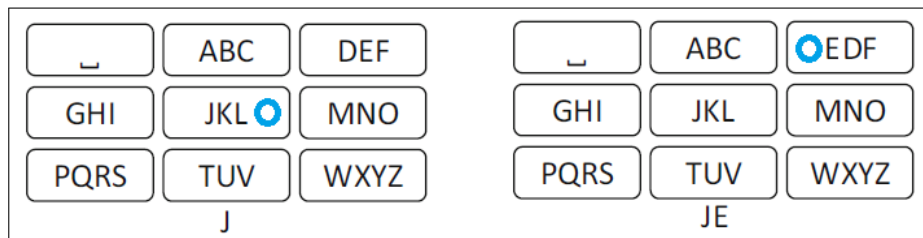


FIGURE 4.31 – Distribution dynamique

Généralement, les méthodes d'entrée de textes combinent les deux distributions. Un exemple est le clavier Dasher [Ward *et al.*, 2000]. Dasher est un clavier utilisable

notamment avec le regard, illustré à la figure 4.32. Dasher positionne toujours les lettres par ordre alphabétique (distribution statique), mais adapte leur taille (distribution dynamique).

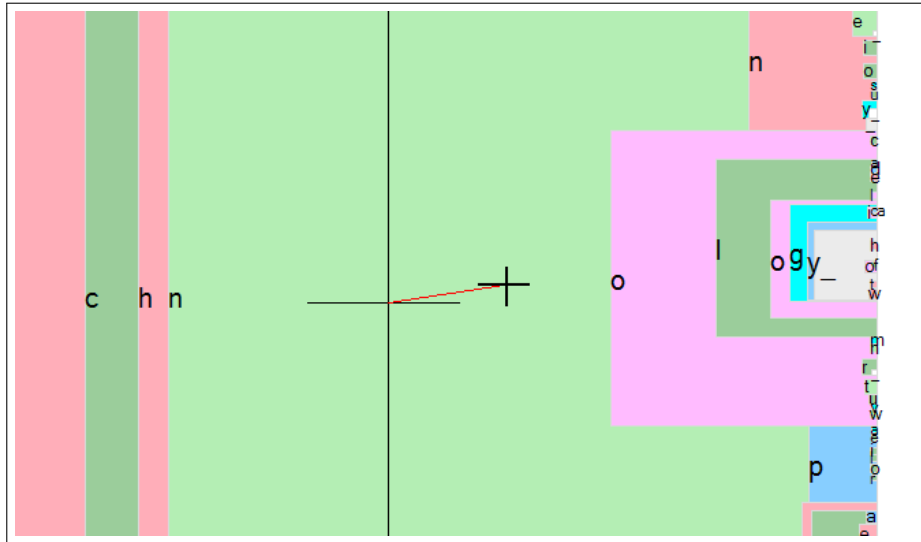


FIGURE 4.32 – Dasher

## 4.4 Conclusion

Ce chapitre a présenté l'état de l'art en matière d'interaction, en abordant les modalités d'interaction (section 4.1) et les moyens d'interaction (sections 4.2 et 4.3).

L'objectif par la suite est de d'abord améliorer l'application VoIP conçue durant le stage en fonction des pistes d'améliorations pointées au chapitre 3, avant d'établir une taxonomie des modalités d'interaction disponibles pour les personnes lourdement handicapées.

# Chapitre 5

## Amélioration de l'interface de téléphonie

Le chapitre 3 soulève une série de pistes pour améliorer l'interface de téléphonie. Ce chapitre propose une nouvelle solution pour répondre au plus possible d'exigences de départ (section 2.1) possible. Le chapitre 4 est utilisé afin de concevoir cette nouvelle solution, en utilisant des méthodes existantes.

L'objectif de ce chapitre est de donner des lignes directrices pour concevoir une application à destination des personnes ayant un handicap moteur, ou un système plus général avec un ensemble d'applications à destination des personnes invalides. Les pistes d'amélioration extraites du chapitre 3 sont les suivantes :

- Utilisation d'outils existants
- Application multi-plateforme
- Meilleure gestion des contacts

L'utilisation d'outils existants est une amélioration qui consiste à être compatible avec le plus possible d'outils d'accessibilités déjà conçus. Le but est de proposer aux utilisateurs cibles la possibilité d'utiliser les outils auxquels ils sont habitués. Un autre avantage est la possibilité pour une entreprise de concevoir ses propres outils d'accessibilité, et de les utiliser sur toutes leurs applications. Ces outils sont les lecteurs d'écrans, les claviers virtuels ainsi que les outils de balayage.

Le développement d'une application multi-plateforme est important, dans le sens où les utilisateurs invalides utilisent des dispositifs différents, en fonction de leurs besoins. Certains possèdent des tablettes Android, d'autres des tablettes Apple, certains des ordinateurs de bureau Windows, etc. Le fait d'avoir une application multi-plateforme permet de répondre aux besoins d'un maximum d'utilisateurs.

Une meilleure gestion des contacts implique une gestion plus centralisée des contacts, dans l'optique notamment d'une utilisation des applications sur plusieurs plateformes différentes. Cela implique également une représentation des données de manière plus universelle, afin d'être compatible avec des carnets d'adresses existants.

La première section détaille la première amélioration possible, c'est-à-dire l'utilisation d'outils d'interaction existants. La deuxième section décrit l'amélioration liée à la gestion des contacts. Ensuite, la section suivante propose l'architecture globale d'une application qui tient compte de ces améliorations. Enfin, la dernière section reprend quelques points sur lesquels il faut faire attention lors de l'implémentation d'une telle solution.

## 5.1 Amélioration des interactions

La première amélioration listée dans le chapitre 3 est la prise en compte d'outils existants. Le chapitre 4 présente toute une série d'outils existants en matière de techniques de balayage. Deux grandes catégories d'applications sont présentées : les applications qui contiennent directement leur outil de balayage, et les outils indépendants.

Les applications qui intègrent un outil de balayage sont des navigateurs web (AVANTI, MultiWeb, Argo, KeySurf et FireScanner). Pour être compatible avec ces outils, il est nécessaire de développer une application basée sur le web.

Afin d'être compatible avec tous les outils indépendants listés dans le chapitre 4, il est nécessaire de concevoir une application compatible avec Mac OS ou Windows. La conception d'une application basée sur le web permet de rencontrer cet objectif.

De plus, une application basée sur le web permet d'utiliser les tags ARIA. Ces tags sont fort utiles notamment dans le cas des personnes aveugles.

Enfin, la conception d'une application web rencontre directement l'amélioration de rendre l'application multi-plateforme.

Dans le cadre de la société Home-Based, une application web rendrait l'application directement intégrable dans Soline, du fait que cette dernière est basée sur les technologies web.

## 5.2 Gestion des contacts

Cette section décrit la manière de gérer l'amélioration concernant les données des contacts. L'objectif est d'avoir une représentation standard pour les contacts, ainsi qu'une gestion centralisée des contacts.

En ce qui concerne la gestion centralisée des contacts, plusieurs options sont disponibles.

La première est la création ou l'utilisation d'un logiciel serveur utilisant le protocole **CardDAV**. Ce protocole est utilisé par les plus grandes entreprises pour la gestion d'un carnet d'adresses. Ces logiciels utilisent **vCard**, une ontologie de représentation des personnes et organisations.

Une deuxième solution est de proposer une connexion aux serveurs de gestion de contacts existants. Par exemple, une solution est de se connecter aux serveurs de Google pour récupérer les contacts liés au compte Google de l'utilisateur. Cela signifie également que, si l'utilisateur possède un téléphone Android, il est possible d'utiliser les contacts de son téléphone.

## 5.3 Architecture globale

Cette section décrit les éléments logiciels qui doivent être présents pour la réalisation d'une application VoIP qui prend en compte toutes les améliorations listées ci-dessus.

L'application web doit contenir trois éléments : un espace de gestion de contacts, une gestion des fonctions VoIP et une gestion des interactions. Ce dernier élément s'occupe également de l'interface graphique de l'application. Pour être utilisable par les personnes aveugles, l'application doit utiliser les tags ARIA. Ces tags permettent également de faciliter l'utilisation des lecteurs d'écrans.

Pour la gestion du protocole expliqué dans la section 2.7, l'application web doit être capable de s'actualiser en fonction des commandes reçues. Pour cela, la mise en place du standard WebSocket est nécessaire. Le standard WebSocket est un protocole réseau qui permet de créer une connexion *full-duplex* sur une connexion TCP. Cela permet d'actualiser une page web sans action de l'utilisateur directement sur cette page web. C'est l'élément de gestion des interactions qui doit s'en occuper.

Toujours pour la gestion du protocole, il faut intégrer un élément central, appelé par la suite **Master**, qui va gérer les dispositifs et transmettre les commandes à l'application web.

L'application web doit avoir un support de **vCard**, afin de gérer les contacts d'un carnet d'adresses **cardDAV**. C'est à l'élément de gestion de contacts que ce rôle revient.

Concernant la téléphonie, il est nécessaire d'avoir un serveur SIP. Une application de téléphonie doit être en permanence allumée afin d'être prête à recevoir un appel. Pour cette raison, il est nécessaire de concevoir un service en arrière-plan, qui se contente d'attendre un appel. Lorsqu'il reçoit un appel, il se doit de lancer l'application web, qui prend alors le relais pour la gestion de l'appel. Ce service doit être installé sur chaque dispositif utilisé par l'utilisateur, afin que, lors d'un appel, l'application web se lance sur tous ces dispositifs. Ainsi, l'utilisateur a la possibilité de répondre à l'appel sur le dispositif de son choix.

La figure 5.1 représente les différents éléments à concevoir, et le liens entre eux. La gestion de l'espace client requiert un serveur CardDAV afin de fonctionner. De même, un dispositif d'entrée (bouton, télécommande) requiert l'élément central du protocole pour contrôler l'application web. Chaque dispositif de sortie (Tablette, Ordinateur, Smartphone) doit intégrer un service SIP qui doit être relié au serveur SIP.

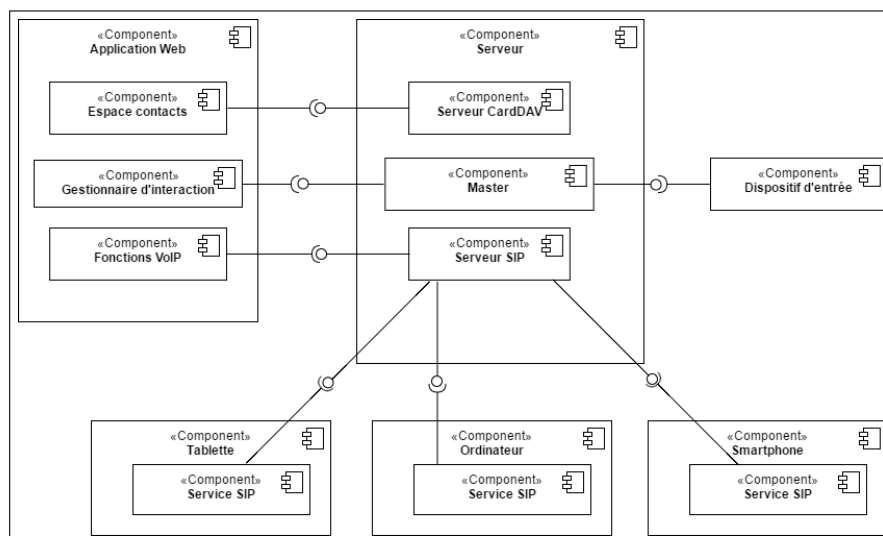


FIGURE 5.1 – Architecture d'une application VoIP



## 5.4 Détails d'implémentation

Cette section explique certains détails d'implémentation des éléments du système.

### 5.4.1 Partie interactions

Cette sous-section reprend les composants suivants : les dispositifs d'entrée, l'élément central du protocole (le **Master**) et le gestionnaire d'interaction.

Lorsque l'utilisateur effectue une action sur un dispositif d'entrée, par exemple un appui sur un bouton, le **Master** intercepte l'évènement d'appui sur le bouton, et transmet ensuite l'information au gestionnaire d'interaction de l'application web, qui exécute la commande liée au bouton.

Le **Master** doit être capable d'intercepter l'évènement qui arrive au système d'exploitation lorsque l'utilisateur appuie sur un bouton. Comme expliqué dans la section 2.7, les modalités d'entrée, hormis la voix, se contentent de simuler un appui sur une touche du clavier. Ce qui signifie que si l'utilisateur possède un bouton configuré pour simuler un appui sur la touche E du clavier, lorsqu'il appuiera sur le bouton, le système d'exploitation recevra la lettre E. Ainsi, dans un éditeur de texte, un appui sur le bouton va afficher la même chose qu'un appui sur la touche E du clavier. Le **Master** doit donc être capable d'identifier de tels évènements. Il doit également être capable de détecter l'origine d'un tel évènement. Si ce dernier provient du clavier, il faut le laisser passer. Sinon, il faut le bloquer et transmettre la commande liée au gestionnaire d'interaction de l'application web. Cette dernière doit d'ailleurs se connecter au **Master** au démarrage de l'application.

La figure 5.2 représente le traitement à effectuer lors de la réception d'un évènement lié aux modalités d'entrée.

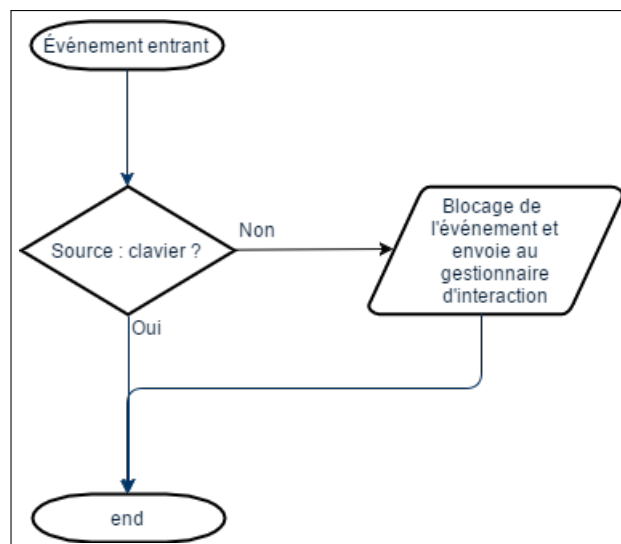


FIGURE 5.2 – Traitement d'un évènement lié aux modalités d'entrée

De plus, le **Master** doit être capable, sur base des modalités d'entrées connectées, de définir le mode d'utilisation de l'application, et le lui transmettre.

Le gestionnaire d'interaction de l'application web doit être en mesure d'adapter l'interface graphique de l'application web en fonction des informations reçues par le

**Master.** Afin de permettre à l'application web et au **Master** de communiquer, il est nécessaire de mettre en place une connexion à l'aide du protocole WebSocket. Ce protocole permet de créer une communication *full-duplex* sur une connexion TCP. C'est par cette connexion que la communication se déroulera.

En ce qui concerne l'interface graphique, étant donné les retours sur celle-ci (section 2.9), garder la même serait une bonne chose.

### 5.4.2 Partie VoIP

Cette sous-section reprend les composants suivants : les fonctions VoIP de l'application web, le serveur SIP ainsi que le service SIP présent sur chaque dispositif de sortie.

L'application devant gérer les fonctions VoIP, il faut trouver une librairie SIP pour du web. La librairie doit être conforme au RFC 7718<sup>1</sup>. Ce RFC décrit les détails techniques de l'utilisation du protocole WebSocket pour le transport d'une connexion SIP. Pour cela, il existe plusieurs solutions existantes, toutes utilisables en Javascript. Notamment JsSIP<sup>2</sup>, un projet open-source écrit par les auteurs du RFC 7718. Deux autres librairies, sipML5<sup>3</sup>, et SIP.js<sup>4</sup>, sont aussi des projets open-source. Ces trois librairies sont conformes au RFC 7718, sont assez bien documentées et sont faciles d'utilisation.

Concernant le serveur SIP, il existe également un bon nombre de solutions. Une bonne solution est OfficeSIP, qui a été utilisée pour l'application conçue durant le stage. Il présente l'avantage d'être léger et facile à configurer.

Le service SIP doit lui se connecter au serveur SIP avec le même identifiant que l'application web. Son rôle est de simplement détecter un appel entrant. Lorsque c'est le cas, il doit ouvrir l'application web. Cela permet, si l'utilisateur possède plusieurs dispositifs de sorties (Tablette, Smartphone, ordinateur, etc), que l'application web s'ouvre sur chacun de ces dispositifs. Cela laisse la possibilité à l'utilisateur de décrocher un appel sur le dispositif de son choix.

### 5.4.3 Partie contacts

Cette sous-section reprend les composants suivants : l'espace des contacts ainsi que le serveur CardDAV.

Comme expliqué dans la section sur les données (5.2), deux solutions sont disponibles. La première est d'utiliser un serveur CardDAV local et la seconde est l'utilisation d'un serveur CardDAV existant.

Pour la première solution, il existe énormément de solutions pour créer un serveur CardDAV. Pour information, il existe le package Fennel<sup>5</sup> pour NodeJS, ou encore Radicale<sup>6</sup>, un serveur CardDAV écrit en python.

- 
1. <https://tools.ietf.org/html/rfc7118>
  2. <http://www.jssip.net/>
  3. <https://www.doubango.org/sipml5/>
  4. <https://sipjs.com/>
  5. <https://github.com/LordEidi/fennel>
  6. <http://radicale.org/>

La deuxième solution requiert une connexion aux serveurs existants. Le plus intéressant à aborder est Google, qui fournit une interface REST<sup>7</sup> pour accéder aux contacts.

Du côté de l'espace client, il faut être capable de récupérer les contacts sur le serveur CardDAV choisi. Il faut donc prendre en charge le format de représentation vCard. De plus, l'ajout, la modification et la suppression d'un contact dans l'application doit se répercuter au niveau du serveur CardDAV.

## 5.5 Analyse des améliorations réalisées

Cette section propose une analyse du système conçu dans ce chapitre.

Le but de ce système était d'améliorer l'application conçue lors du stage (chapitre 2), en tenant compte des améliorations possibles identifiées au chapitre 3. Bien évidemment, ce système devait répondre aux exigences déjà remplies par l'application développée durant le stage.

Le système conçu ici répond bien à toutes ces exigences. En effet, il est possible, grâce à ce système, de passer un appel, de recevoir un appel, d'envoyer un code DTMF lors d'un appel, etc. De plus, ce système est compatible avec une multitude de modalités d'interaction. Étant basé sur le web, le système est également compatible avec une grande partie des systèmes d'exploitations existants, moyennant le développement d'un service VoIP léger pour chaque système d'exploitation. De plus, le fait que l'application soit basée sur le web permet de la rendre automatiquement compatible avec les moyens d'interaction abordés à la section 4.2.2, c'est-à-dire les outils de balayage. Le système est également adapté aux personnes aveugles, grâce à la prise en charge des tags ARIA.

Un gros avantage de ce système par rapport à l'application conçue initialement est sa capacité d'être adaptée à un autre domaine que la téléphonie. En effet, plusieurs composants sont généraux, notamment tout ce qui concerne les interactions. Il suffirait simplement de ré-utiliser ces composants généraux dans la conception d'une application appliquée à un autre domaine.

---

7. <https://developers.google.com/google-apps/carddav/>

## Chapitre 6

# Élaboration d'une taxonomie des modalités d'interaction

Ce chapitre présente l'élaboration d'une taxonomie des modalités d'interaction, afin d'aider à choisir la ou les modalités à utiliser en fonction des difficultés des utilisateurs. Cette taxonomie se base sur les modalités présentée dans l'état de l'art, au chapitre 4, ainsi que sur les conseils de Home-Based et du CRETH (section 2.3).

Afin de choisir la modalité la plus adaptée à la personne, il convient d'identifier les difficultés que peuvent rencontrer les utilisateurs. Il faut également prendre en compte l'aspect de facilité d'utilisation des modalités. En effet, si l'utilisateur peut facilement utiliser ses mains, l'usage de boutons est plus aisée que l'usage du regard, qui demande une concentration extrême et qui amène rapidement une grosse fatigue.

La première question à se poser est la capacité de l'utilisateur à utiliser ses mains. S'il a une mobilité aisée de ses mains, il peut utiliser les interrupteurs boutons, le joystick, l'interrupteur à levier ainsi que l'interrupteur à feuille.

Dans le cas où l'utilisateur n'a pas la capacité de bouger ses mains facilement, trois choix s'offrent à lui. Soit il est capable de serrer un objet et peut donc utiliser l'interrupteur à poigne ou à pression d'air. Soit il peut bouger légèrement ses doigts, cas où il peut utiliser les interrupteurs à pouce, à doigt et à ficelle. Enfin, s'il ne peut bouger ses doigts mais peut légèrement bouger ses mains, il peut utiliser le capteur de proximité ou l'interrupteur infrarouge.

Si l'utilisateur n'est pas capable de se servir de ses mains, quatre autres choix sont disponibles : l'usage des pieds (interrupteur à pédale), l'usage de la tête (interrupteur à ruban ou au menton) l'usage de la respiration (interrupteur pneumatique) ou l'usage de la mâchoire (interrupteur à mordre).

Toutes les modalités citées ci-dessus sont des modalités mécaniques. Si aucune ne convient à l'utilisateur, d'autres pistes sont envisageables.

La première piste à explorer est l'usage de la parole. Cependant, il faut une voix reconnaissable par les systèmes informatiques. En effet, il arrive que certaines voix, faibles ou encore enrouée, ne soient pas reconnues.

Une autre piste est l'usage du regard. Il faut cependant prendre en compte la difficulté d'utilisation du regard au quotidien. En effet, l'usage du regard requiert une importante concentration. Cela amène également une grande fatigue après une

courte période d'utilisation. Cette modalité n'est d'ailleurs utilisée que dans les cas où aucune autre n'est disponible.

L'arbre représenté dans la figure 6.1 est une conclusion de ce qui a été dit. Il ne concerne que les modalités mécaniques.

Il s'agit ici d'une première approche dans l'élaboration d'une taxonomie des modalités d'interaction. Cette première approche produit déjà un résultat utilisable, mais ouvre la porte à plusieurs améliorations possibles.

La première serait de faire valider scientifiquement cette première taxonomie. En effet, cela permettrait de donner un peu de poids à celle-ci. La validation peut se faire par exemple en interrogeant des experts du domaine comme des ergothérapeutes, des médecins, etc. Une autre solution, plus compliquée à mettre en place, consisterait à tester cette taxonomie sur un large panel d'utilisateurs, aux handicaps différents. L'idée serait de faire essayer les différentes modalités proposées par cette taxonomie en fonction du handicap de la personne, et vérifier si l'utilisation des modalités est possible pour l'utilisateur.

Une deuxième amélioration serait d'étoffer cette taxonomie en prenant en compte notamment la complexité d'utilisation ou le niveau d'intrusion d'une modalité. Cela permettrait de proposer directement une modalité qui serait la plus optimale possible, en fonction des difficultés rencontrés par la personne.



FIGURE 6.1 – Arbre de décision des modalités mécaniques à utiliser



# Chapitre 7

## Conclusion et travaux futurs

Les deux questions de recherche de ce travail étaient les suivantes :

1. Comment permettre aux personnes lourdement handicapées de pouvoir téléphoner ainsi que d'ouvrir la porte d'entrée via un portique aisément ?
2. Quelles modalités d'interaction sont-elles disponibles en fonction du handicap de la personne ?

Afin d'atteindre ces objectifs, plusieurs étapes ont été nécessaires :

En premier lieu, il a fallu concevoir une première application VoIP. Cela s'est fait en immersion, dans le cadre du stage, au sein de l'entreprise Home-Based. Le résultat de l'étape de l'immersion chez Home-Based est une solution VoIP fonctionnelle, qui répond aux exigences initiales. C'est-à-dire, une application adaptée aux personnes lourdement handicapées, qui intègre les moyens d'interaction les plus courants pour ces utilisateurs. L'application est contrôlable par des boutons poussoirs, par une télécommande, par une souris, par un clavier, etc.

Afin de concevoir cette application, plusieurs rencontres ont eu lieu avec plusieurs experts du domaine des interfaces et des interactions pour les personnes lourdement handicapées. C'est-à-dire les fondateurs de Home-Based ainsi que plusieurs ergothérapeutes. Cela a permis d'avoir toute une série d'information sur les modalités et moyens d'interaction utilisée par les personnes lourdement handicapées.

Dans le but de valider une première version du système développé, une rencontre avec une utilisatrice a eu lieu.

Par la suite, un regard critique a été posé sur le système développé, en identifiant ses forces ainsi que ses améliorations possibles.

Dans le but de répondre à ces améliorations, une phase de recherche a été réalisée. La cible de ces recherches était les modalités et les moyens d'interaction existantes pour les personnes invalides.

Ensuite, l'étape suivante était d'améliorer le système, en tenant compte des améliorations identifiées et du résultat des recherches.

Du point de vue des exigences de l'application, celle-ci devait dans un premier temps avoir une interface graphique adaptée aux personnes lourdement handicapées. Afin d'interagir avec l'application, l'utilisateur devait pouvoir utiliser une grande variété de modalités d'entrée, notamment des boutons poussoirs, une télécommande,



etc. D'un point de vue fonctionnel, les fonctions VoIP devaient être implémentées, tout comme une gestion des contacts et des historiques d'appels. Enfin, l'application devait être, au minimum, compatible avec Windows.

Le système conçu dans ce travail répond bien à toutes ces exigences. De plus, étant basé sur le web, le système est compatible avec Windows, Linux, Mac OS, Android, Windows Phone et iOS pour ne citer que les principaux. Ceci moyennant le développement d'un service VoIP minimaliste pour chaque système d'exploitation. Le système est également adapté aux personnes aveugles, grâce à la prise en charge des tags ARIA.

Un avantage du système conçu est qu'il peut facilement être adapté à un autre domaine que la téléphonie. Plusieurs composants sont généraux, notamment tout ce qui concerne les interactions.

Ces étapes permettent d'apporter une réponse à la première question de recherche, dans le sens où le système permet aux personnes lourdement handicapées de téléphoner et d'ouvrir une porte d'entrée. De plus, l'utilisatrice interviewée utilise toujours cette application à l'heure actuelle. Ce qui montre son utilité dans un cas concret.

Enfin, à la suite des recherches effectuées lors de l'état de l'art, il a été possible d'établir une taxonomie des modalités d'interaction, afin d'aider à choisir la ou les modalités à utiliser en fonction des difficultés des utilisateurs. Cette étape a permis de répondre à la deuxième question, qui était de choisir une modalité adaptée à l'utilisateur en fonction de son handicap. Le résultat de cette question est un arbre de décision qui permet, en fonction des capacités de l'utilisateur, d'orienter le choix d'une modalité vers une adaptée.

## 7.1 Travaux futurs

Pour la suite, plusieurs pistes sont envisageables. La première serait de faire tester l'interface et les interactions de l'application à un panel plus large d'utilisateur. Il serait intéressant de trouver plusieurs personnes avec des handicaps différents, afin de voir si l'application est utilisable par un maximum de personnes atteintes de handicaps différents.

Une deuxième piste serait une intégration plus poussée avec le système Soline, en intégrant notamment le TTS.

Ensuite, il serait intéressant de concevoir une application à partir du système conçu dans ce travail, dans un autre domaine que la téléphonie. Cela permettrait de tester le système dans un autre domaine.

Une étude de comparaison des différentes techniques de balayages rencontrées lors des recherches serait une dernière piste à explorer. Cela permettrait de, tout comme les modalités d'entrée, établir quelle est la technique ou quelles sont les techniques privilégiées en fonction du handicap de la personne.

Concernant la taxonomie des modalités, une tâche à réaliser serait de la faire valider scientifiquement, auprès notamment d'experts du domaine des personnes lourdement handicapées.

Enfin, l'élaboration d'une taxonomie des modalités plus riche encore serait envisageable. Elle pourrait notamment prendre en compte la complexité d'utilisation d'une modalité ou encore le niveau d'intrusion d'une modalité.



# Bibliographie

- [Betke *et al.*, 2002] BETKE, M., GIPS, J. et FLEMING, P. (2002). The camera mouse : visual tracking of body features to provide computer access for people with severe disabilities. *IEEE Transactions on neural systems and Rehabilitation Engineering*, 10(1):1–10.
- [Biswas et Robinson, 2008] BISWAS, P. et ROBINSON, P. (2008). A new screen scanning system based on clustering screen objects. *Journal of Assistive Technologies*, 2(3):24–31.
- [Boissiere et Dours, 2003] BOISSIERE, P. et DOURS, D. (2003). An overview of existing writing assistance systems. *In Proceedings of the IFRATH Workshop*.
- [Bolt, 1980] BOLT, R. A. (1980). “Put-that-there” : Voice and gesture at the graphics interface, volume 14. ACM.
- [Cooper et Associés, 2012] COOPER, R. et ASSOCIÉS (2012). Crossscanner : 1-2 switch mouse emulator. <http://www.rjcooper.com/cross-scanner/index.html>. Accès : 12-03-2017.
- [Dumas *et al.*, 2009] DUMAS, B., LALANNE, D. et OVIATT, S. (2009). Multimodal interfaces : A survey of principles, models and frameworks. *Human machine interaction*, pages 3–26.
- [Factors, 2017] FACTORS, A. H. (2017). Scanbuddy. <http://www.orin.com/access/switchxs/>. Accès : 12-04-2017.
- [Felzer et Freisleben, 2002] FELZER, T. et FREISLEBEN, B. (2002). Hawcos : the hands-free wheelchair control system. *In Proceedings of the fifth international ACM conference on Assistive technologies*, pages 127–134. ACM.
- [Instruments, 2017] INSTRUMENTS, O. (2017). Switchxs. <http://newsite.ahf-net.com/scanbuddy/>. Accès : 12-04-2017.
- [Jacob, 1991] JACOB, R. J. (1991). The use of eye movements in human-computer interaction techniques : what you look at is what you get. *ACM Transactions on Information Systems (TOIS)*, 9(2):152–169.
- [Ntoa *et al.*, 2013] NTOA, S., MARGETIS, G., ANTONA, M. et STEPHANIDIS, C. (2013). Scanning-based interaction techniques for motor impaired users. *Assistive Technologies and Computer Access for Motor Disabilities*, G. Kouroupetroglou, Ed. IGI Global.
- [Ntoa *et al.*, 2009] NTOA, S., MARGETIS, G. et STEPHANIDIS, C. (2009). Firescanner : A browser scanning add-on for users with motor impairments. *In International Conference on Universal Access in Human-Computer Interaction*, pages 755–763. Springer.

- [Ntoa *et al.*, 2004] NTOA, S., SAVIDIS, A. et STEPHANIDIS, C. (2004). Fastscanner : An accessibility tool for motor impaired users. *In International Conference on Computers for Handicapped Persons*, pages 796–803. Springer.
- [Ntoa et Stephanidis, 2005] NTOA, S. et STEPHANIDIS, C. (2005). Argo : A system for accessible navigation in the world wide web. *ERCIM News*, 61:53–54.
- [Owens et Keller, 2000] OWENS, J. et KELLER, S. (2000). *Multiweb : Australian contribution to web accessibility*. Institute of Disability Studies, School of Management Information Systems.
- [Polacek *et al.*, 2015] POLACEK, O., SPORKA, A. J. et SLAVIK, P. (2015). Text input for motor-impaired people. *Universal Access in the Information Society*, pages 1–22.
- [Savidis *et al.*, 1997] SAVIDIS, A., VERNARDOS, G. et STEPHANIDIS, C. (1997). Embedding scanning techniques accessible to motor-impaired users in the windows object library. *Advances in human factors/ergonomics*, pages 429–432.
- [Sears *et al.*, 2001] SEARS, A., KARAT, C.-M., OSEITUTU, K., KARIMULLAH, A. et FENG, J. (2001). Productivity, satisfaction, and interaction strategies of individuals with spinal cord injuries and traditional users interacting with speech recognition software. *Universal Access in the information Society*, 1(1):4–15.
- [Simpson *et al.*, 2006] SIMPSON, R., KOESTER, H. et LOPRESTI, E. (2006). Evaluation of an adaptive row/column scanning system. *Technology and disability*, 18(3):127–138.
- [Simpson et Koester, 1999] SIMPSON, R. C. et KOESTER, H. H. (1999). Adaptive one-switch row-column scanning. *IEEE Transactions on Rehabilitation Engineering*, 7(4):464–473.
- [Spalteholz *et al.*, 2008] SPALTEHOLZ, L., LI, K. F., LIVINGSTON, N. et HAMIDI, F. (2008). Keysurf : A character controlled browser for people with physical disabilities. *In Proceedings of the 17th international conference on World Wide Web*, pages 31–40. ACM.
- [Stabel, 2013] STABEL (2013). Indicateurs démographiques de la belgique 2016-2060. [http://statbel.fgov.be/fr/binaries/CoefBelgFR\\_tcm326-282319.xls](http://statbel.fgov.be/fr/binaries/CoefBelgFR_tcm326-282319.xls). Accès : 15-03-2017.
- [Stephanidis *et al.*, 1998] STEPHANIDIS, C., PARAMYTHIS, A., SFYRAKIS, M., STERGIOU, A., MAOU, N., LEVENTIS, A., PAPAROULIS, G. et KARAGIANNIDIS, C. (1998). Adaptable and adaptive user interfaces for disabled users in the avanti project. *Intelligence in Services and Networks : Technology for Ubiquitous Telecom Services*, pages 153–166.
- [Stephanidis et Savidis, 2002] STEPHANIDIS, C. et SAVIDIS, A. (2002). Unified user interface development. *In The human-computer interaction handbook*, pages 1069–1089. L. Erlbaum Associates Inc.
- [Steriadis et Constantinou, 2002] STERIADIS, C. E. et CONSTANTINOU, P. (2002). Using the scanning technique to make an ordinary operating system accessible to motor-impaired users. the “autonomia” system. *Group*, 8(B7):B6.
- [Steriadis et Constantinou, 2003] STERIADIS, C. E. et CONSTANTINOU, P. (2003). Designing human-computer interfaces for quadriplegic people. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 10(2):87–118.

- [Ward *et al.*, 2000] WARD, D. J., BLACKWELL, A. F. et MACKAY, D. J. (2000). Dasher—a data entry interface using continuous gestures and language models. *In Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 129–137. ACM.
- [Zhao *et al.*, 2012] ZHAO, X. A., GUESTRIN, E. D., SAYENKO, D., SIMPSON, T., GAUTHIER, M. et POPOVIC, M. R. (2012). Typing with eye-gaze and tooth-clicks. *In Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 341–344. ACM.